

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

August 2022

Issue #57

Robot Roundup

Make your Raspberry Pi move

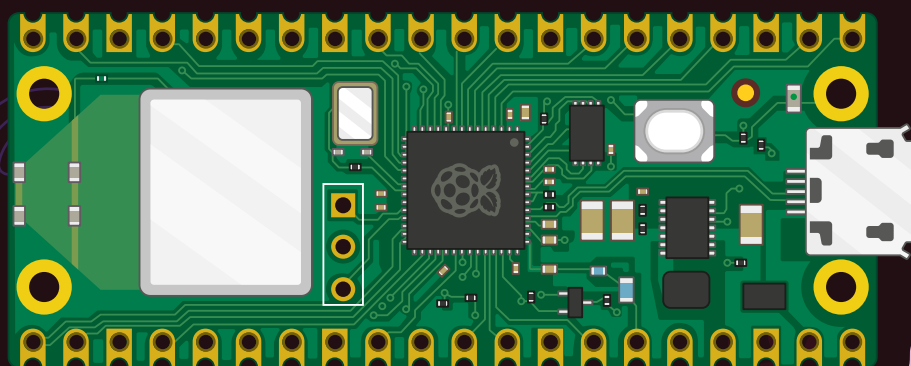
RASPBERRY PI

Beer BBQ

Turn a keg into an outdoor grill

PICOW

PROJECTS



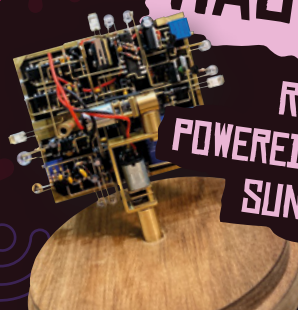
Make internet-connected builds with the **NEW** \$6 microcontroller board

Time-lapse Photography

Watch life on fast forward

BEAM MACHINE

ROBOTICS
POWERED BY THE
SUN'S RAYS



LASER CUTTING KNITTING WELDING DOGS

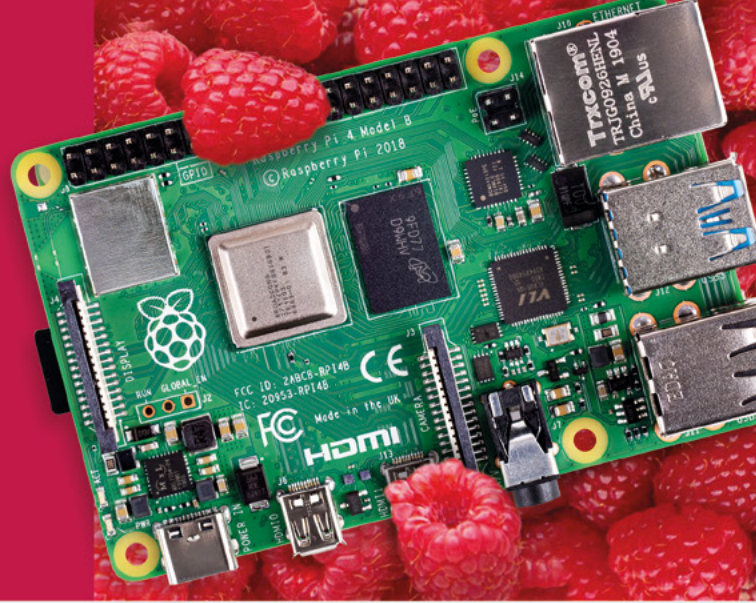


Aug. 2022
Issue #57 £6

VISIT...

LANZAROTE
Caliente.COM

American Raspberry Pi Shop



- Displays
- Cases
- Project Kits
- Add-on Boards
- HATs
- Arcade
- Cameras
- Cables and Connectors
- Sensors
- Swag
- Power Options
- GPIO and Prototyping

Partner and official reseller for top Pi brands:



and many
others!

Price, service, design,
and logistics support for
VOLUME PROJECTS

USA

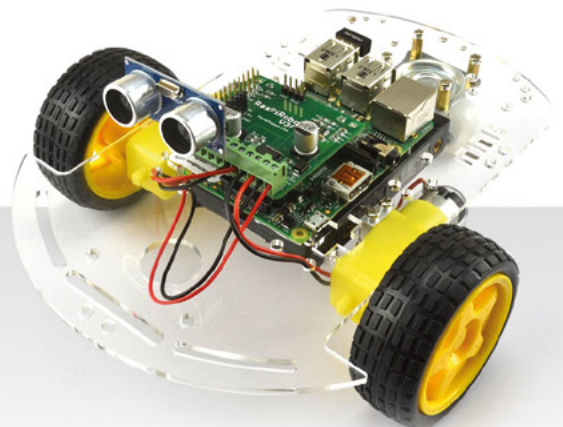


PiShop.us

Canada



PiShop.ca



Raspberry Pi

APPROVED RESELLER



Welcome to HackSpace magazine

When Pico came out in January 2021, there was one feature requested more than any other – wireless networking. The problem is, while there was some spare space on the Pico board, there wasn't really enough for wireless. There also weren't any spare GPIOs that could be used to connect the wireless controller. These two obstacles didn't stop the Raspberry Pi engineers, who ploughed on regardless.

This month, we're putting the Pico W through its paces with three projects to get you started on your own connected projects

The result of their hard work is the Pico W. It's (almost) pin and form factor compatible with Pico – the only changes

being moving the debug pins and a change to the LED. This month, we're putting the Pico W through its paces with three projects to get you started on your own connected projects. Whether you want to build an IoT device, control a robot, or just flash some lights, we can help.

BEN EVERARD

Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.com

[f hackspacemag](#)

[hackspace](#)

ONLINE

hsmag.cc



PAGE 44
FREE PICO
WHEN YOU
SUBSCRIBE

EDITORIAL

Editor

Ben Everard

ben.everard@raspberrypi.com

Features Editor

Andrew Gregory

andrew.gregory@raspberrypi.com

Sub-Editors

David Higgs, Nicola King

DESIGN

Critical Media

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Sam Ribbits, Olivia Mitchell, Ty Logan

Photography

Brian O'Halloran

CONTRIBUTORS

Jo Hinchliffe, Marc de Vinck, Andrew Lewis, Mike Bedford, Nicola King, Rosie Hattersley, Jack Spiggle, Rob Miles

PUBLISHING

Publishing Director

Russell Barnes

russell@raspberrypi.com

Advertising

Charlie Milligan

charlotte.milligan@raspberrypi.com

DISTRIBUTION

Seymour Distribution Ltd

2 East Poultry Ave,

London EC1A 9PT

+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre, Kelvin Lane, Manor Royal, Crawley, West Sussex, RH10 9PE

To subscribe

01293 312189

hsmag.cc/subscribe

Subscription queries

hackspace@subscriptionhelpline.co.uk

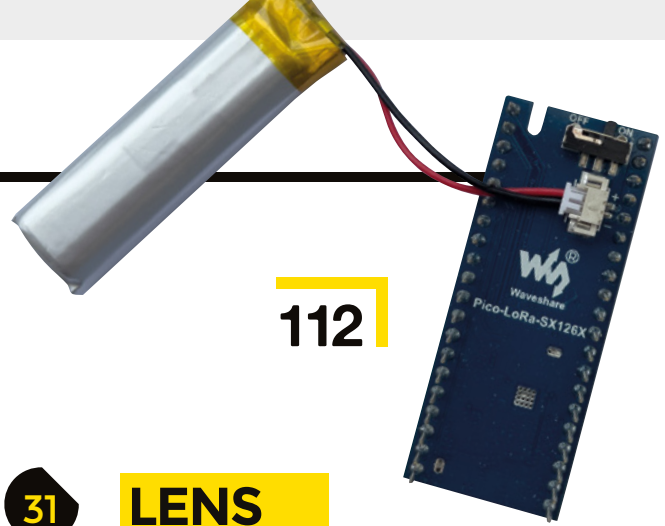


This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents

112



06

SPARK

- 06 Top Projects**
Beautiful builds made by humans
- 18 Objet 3d'art**
It's amazing what you can do with hot plastic
- 20 Meet the Makers: EMF Camp**
Our report from the Glastonbury of makers
- 26 Letters**
People love rockets. Fair enough...
- 28 Crowdfunding**
A tiny Linux computer that Prince would have loved

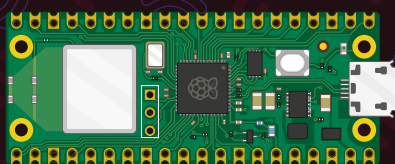
31

LENS

- 32 Pico W Projects**
Build connected projects with the new board from Raspberry Pi
- 46 How I Made: Solar tracking head**
Free-form soldering meets solar power in a BEAM robot
- 52 Interview: Andrew Sink**
Photogrammetry, printing, and all things 3D
- 60 Improviser's Toolbox Barbed wire**
From the spiky came forth sweetness
- 64 In the workshop Beer keg BBQ**
A metalwork project that combines two of our favourite things

Cover Feature

PICO W PROJECTS



Make internet-connected builds with the **NEW \$6** microcontroller board

32

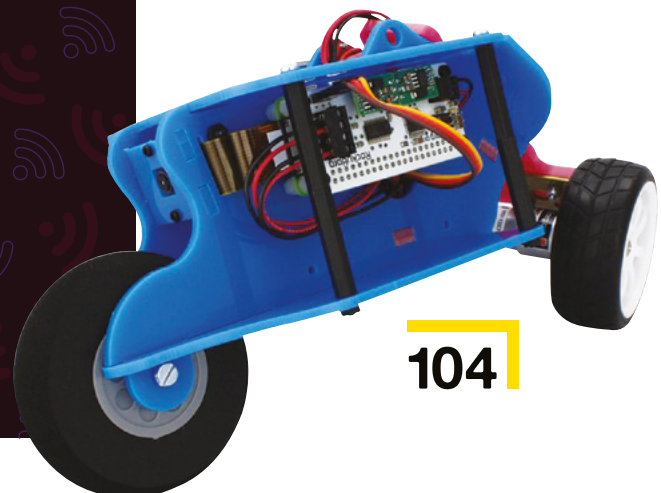
Tutorial

Knitting



78

Turn programmatically stored information into physical patterns



104

96

How I Made BEAM robotics



46

Create a very simple AI with brass and capacitors

69

FORGE

70

SoM Welding helmet

Pimp your PPE with budget-friendly add-ons

74

SoM Raspberry Pi

Time-lapse photography on a Raspberry Pi

78

Tutorial Knitting

Knitting: it's like coding, but with fabric

84

Tutorial Laser cutter

How to calibrate your newly improved K40

90

Tutorial Infrared photography

Another way to enjoy the wonders of nature

96

Tutorial Printing camera

Relive the glory days of Polaroid, with Python

18



06

20



Interview

Andrew Sink



52

That's one way to defeat face-recognition technology



103

FIELD TEST

70

104

Best of Breed

The best robotics add-ons for Raspberry Pi

112

Review Waveshare SX1262

Add LoRa capabilities to a Raspberry Pi Pico

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits. HackSpace magazine is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US. Application to mail at Periodicals prices is pending at Williamsport, PA. Postmaster please send address changes to HackSpace magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

(Freddie) Mercury thermometer

By TurboSnail

hsmag.cc/FreddieMercuryThermometer

Has there ever been a more captivating rock front man than **Freddie Mercury**? No, of course there hasn't. That's what makes this custom temperature display so awesome. Well, that and the custom PCB that forms the display, and the custom microcontroller based on an Arduino Uno that forms the guts of the build (along with a low-power servo and an Adafruit AHT20 temperature/humidity sensor).

All together now: DAAAA-OOOOOOOOH! 

**Right **

Freddie points at the temperature with his resin-printed arm



RFID Record Player

By Ibrahima, Scott, Antoine & Arthur

hsmag.cc/RFID_RecordPlayer

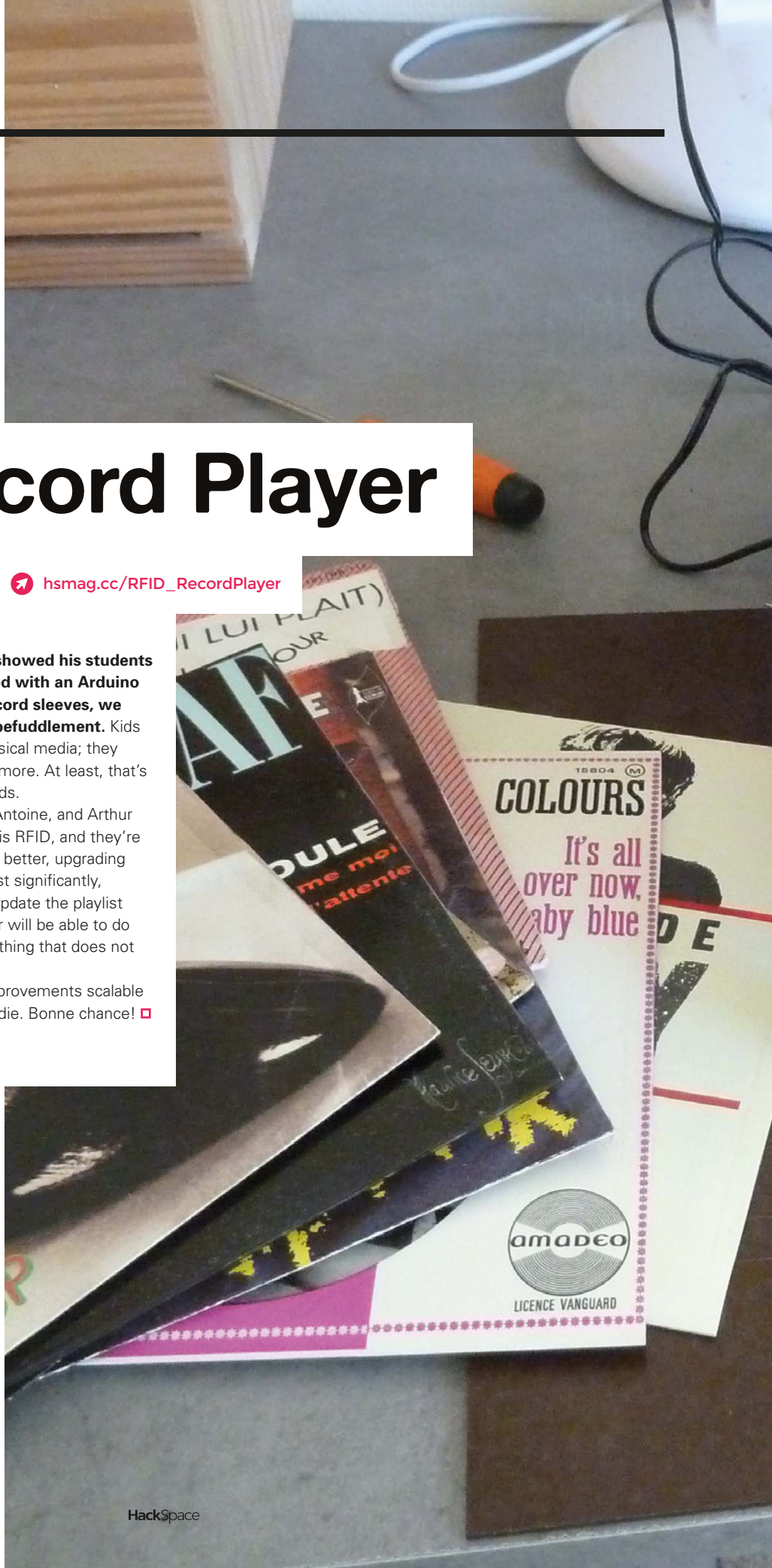
When teacher Jean Noël showed his students a record player upgraded with an Arduino to read RFID tags on record sleeves, we can only imagine their befuddlement. Kids these days don't use physical media; they don't even download anymore. At least, that's what we're told. We don't actually know any kids.

The students in question – Ibrahima, Scott, Antoine, and Arthur – did recognise that the underlying technology is RFID, and they're working on making their teacher's project even better, upgrading the user interface to make it easier to use. Most significantly, instead of uploading a new Arduino sketch to update the playlist (something that requires coding skills), the user will be able to do it by renaming MP3 files on the SD card (something that does not require coding skills).

The students eventually plan to make the improvements scalable as a kit form that they will be able to sell on Tindie. Bonne chance! ▣

Right ▣

Other improvements in the pipeline include an RGB LCD screen to give information to the user and show some cool light animations





Mini Router Plane

By WOmadeOD

hsmag.cc/RouterPlane

W

orkshop Heaven, the online emporium of woodworking delights, has listed among its wares a hand plane going for £10,560.

It's a beautiful tool, but is it more beautiful than this handmade router plane built out of scraps of wood in the maker's workshop?

We'd say not. This build, by Instructables user WOmadeOD comprises a chunk of hardwood, a square of steel plate, a nut, and a bolt, which is shaped, hardened, and sharpened until it can take off thin slivers of wood. ▣

Right ▣

To adjust the depth of the plane, just turn the bolt a fraction of a turn



Raspberry Pi TV simulator

By Pakéquis

 Pakequis.com.br

In this age of LED everything, it's hard to imagine the tiny CRT televisions that were once marketed as 'travel' TVs. That's what this is, only this has been retrofitted to include a Raspberry Pi playing a selection of old black-and-white films. Even better, when you change the channel, it produces an authentic screen of white noise, just like TVs used to do. 



Right  Movies are stored on the Raspberry Pi's SD card, and there's an Arduino Pro Mini to provide voltage that the TV can use



3D-printed bike light

By Kevr102

hsmag.cc/BikeLight

After ‘do not crash’, our one golden rule of riding a bike is that you can never have too many lights. That’s what inspired Kevr102 to design and build this auxiliary bike light for when his existing front light runs out of battery. It’s a 3D-printed enclosure for a Kitronik 5V LED lamp kit, with the addition of a buck converter to drop the 6V from the AA batteries down to the 5V required by the LEDs. And as the maker notes, the square version “looks a bit like the weapon the Predator has on its shoulder”. ▣

Right ▣
Kevr102 designed this bike light in Fusion 360





3D-printed power loom

By Fraens

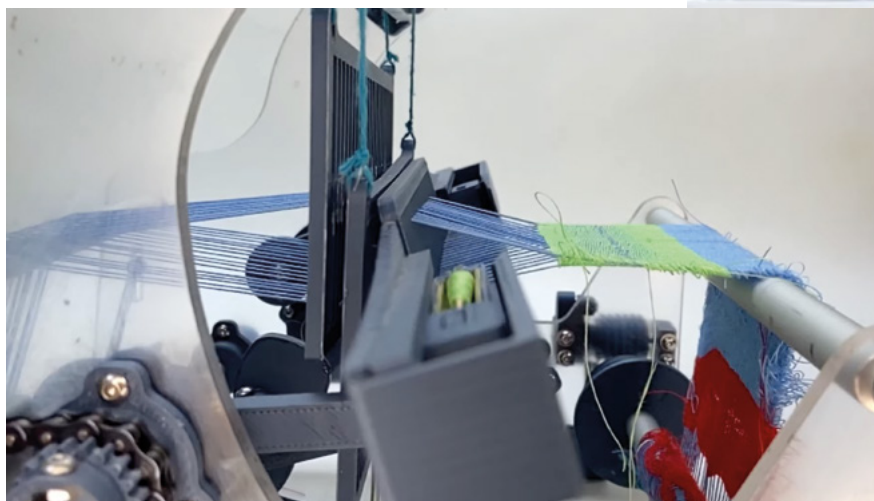
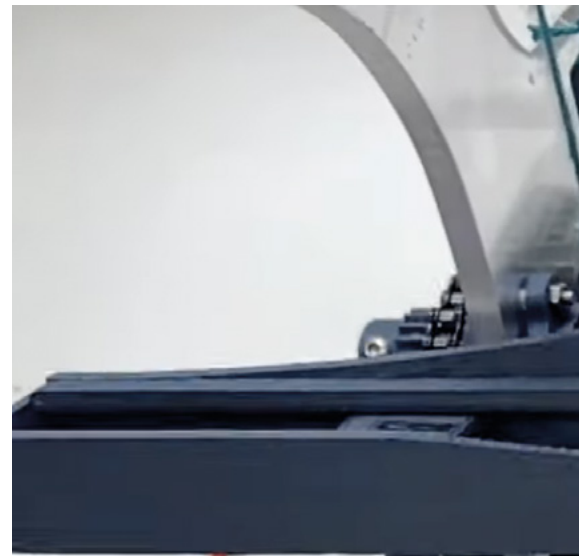
 hsmag.cc/PowerLoom

This issue, we've gone into the coding behind knitting and crochet, and raised a hat to the Jacquard loom, which took instructions that could've been programmed anywhere, by anyone. It

wasn't Turing-complete, but it was, almost, a computer.

This 3D-printed model pays tribute to the power looms that laid the way for the Industrial Revolution and, indirectly, computer science.

Fraens has made the STL files available on Printables and Thingiverse, and there's a full list of materials used on his YouTube page, including screws, ball bearings, and a 12 V motor. Apparently, one of the hardest things to get right was the weight of the shuttle: too heavy and it won't move; too light and it gets snagged easily. 



Right 
The Luddites would never have smashed a machine they could have printed themselves



Objet 3d'art

3D-printed artwork to bring more beauty into your life

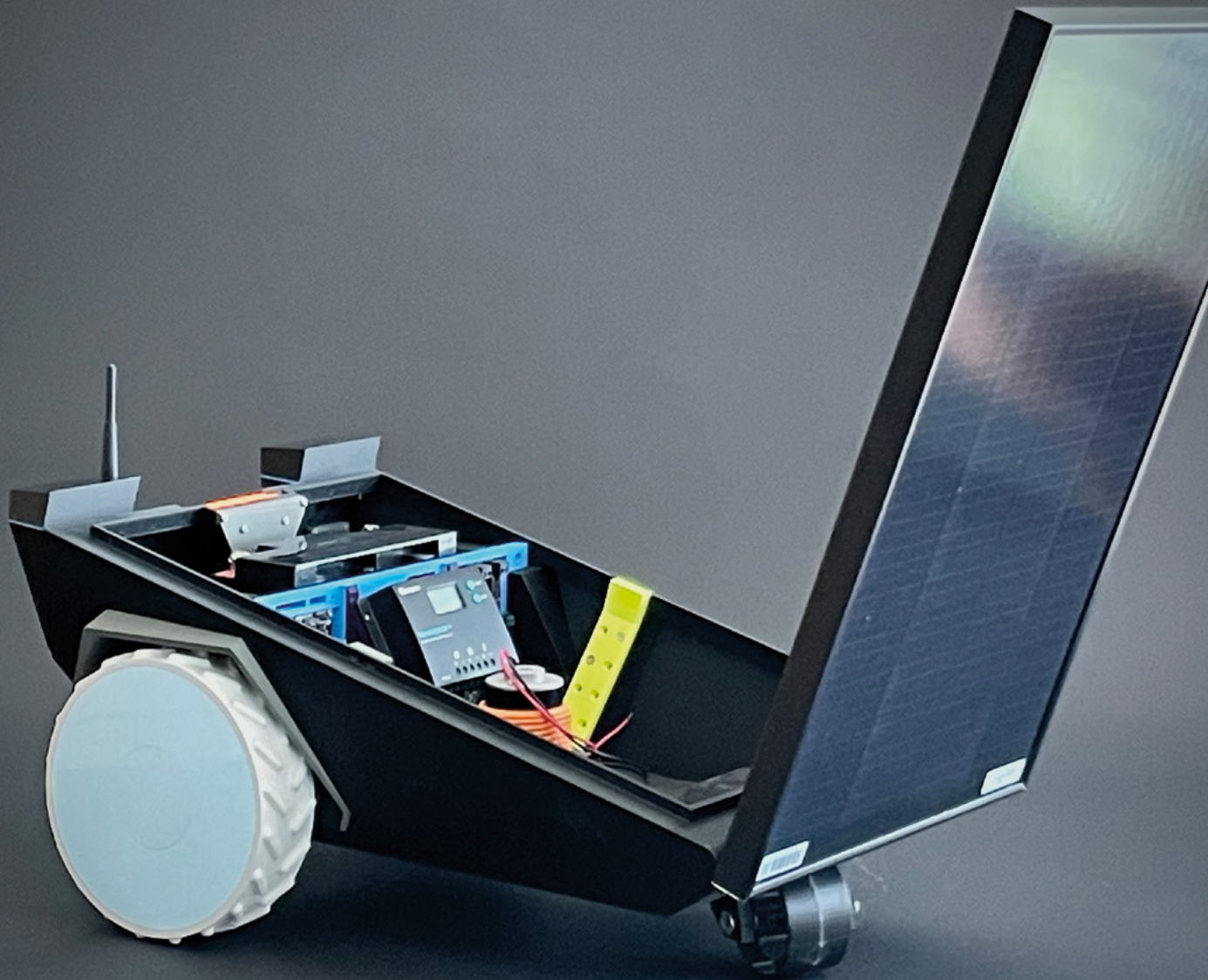


Most of the time, it's a pleasure to mow one's lawn. It's a moment of communing with the outdoors, of exerting your mastery over nature.

Mowing the lawn is a lovely job – unless you're stricken with hay fever. Or, you have extensive grounds that you just can't be bothered to look after (admittedly, this would be a nice problem to have).

In these situations, a robotic mower comes into its own – and TGD Consulting has come up with a 3D-printed, Raspberry Pi-controlled solution: PiMowBot. Like the BEAM project featured on page 46, it takes power from the sun, so the whole top panel is a solar panel. You can get the design files from Cults3D below. [■](#)

hsmag.cc/PiMowBot



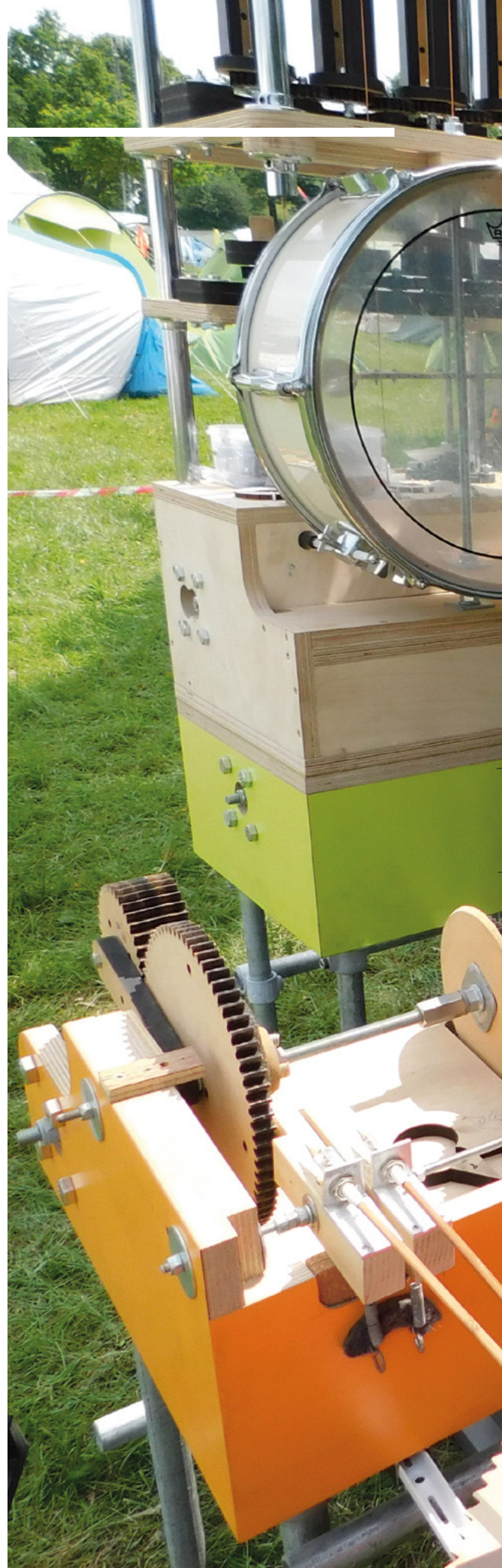
Electromagnetic Field 2022

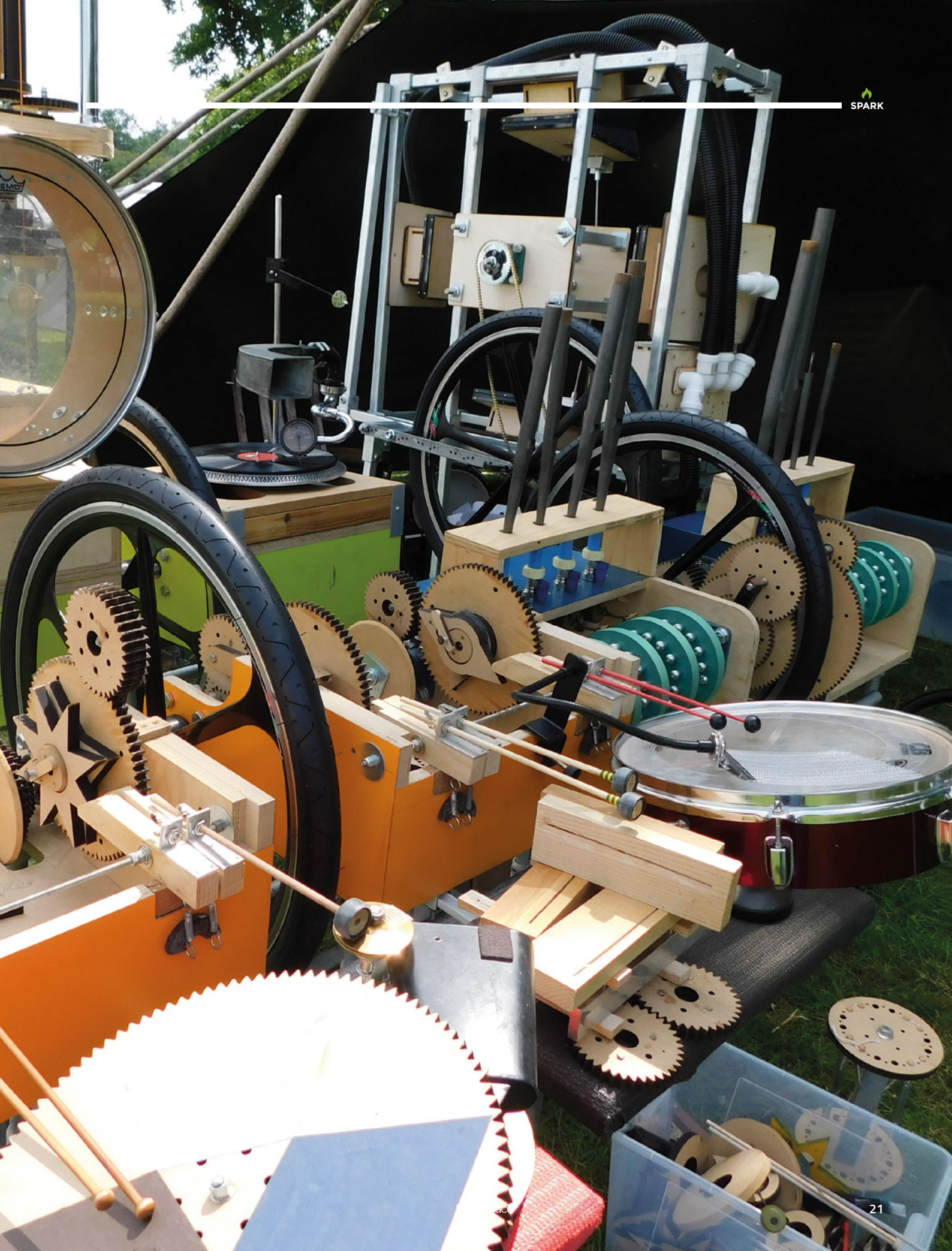
Our roving reporter visits the UK's biannual maker festival

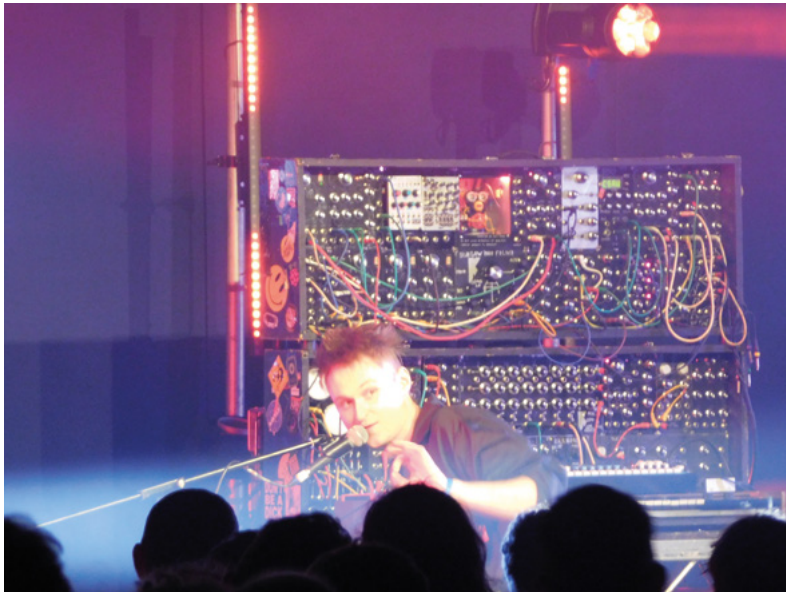
Every two years, the British summer is lit up by **Electromagnetic Field (also known as EMF Camp)**. This maker festival brings people together to show off makes and talk about the things they've been doing. Since 2018, this has been held in Eastnor Castle just outside Ledbury in Herefordshire (a county forever blighted by Robert De Niro's mispronunciation in the film *Ronin*: for our international audience, it's pronounced approximately hair-re-ford-shire).

" For some, it's about how they built a particular project; **for others, it's about a technology they've been working with**

The bulk of the festival is taken up by talks. Across three stages, people from the makersphere give talks about almost any topic. For some, it's about how they built a particular project; for others, it's about a technology they've been working with. Others are, well, about almost anything. To give you a bit of a flavour, here are some that we went to see: →







Above ♦
Sam Battle valiantly
pressing on
through some
technical hiccups



Solarpunks assemble

Terence Eden talked us through his experiences running domestic solar setups on three of the houses he's lived in, what's worked well, and why he thinks we should all be running domestic solar setups.

Wearable fire art

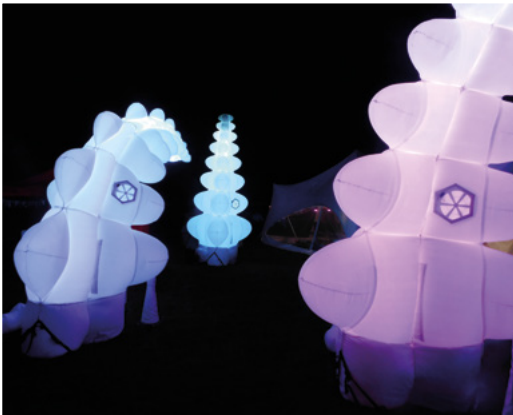
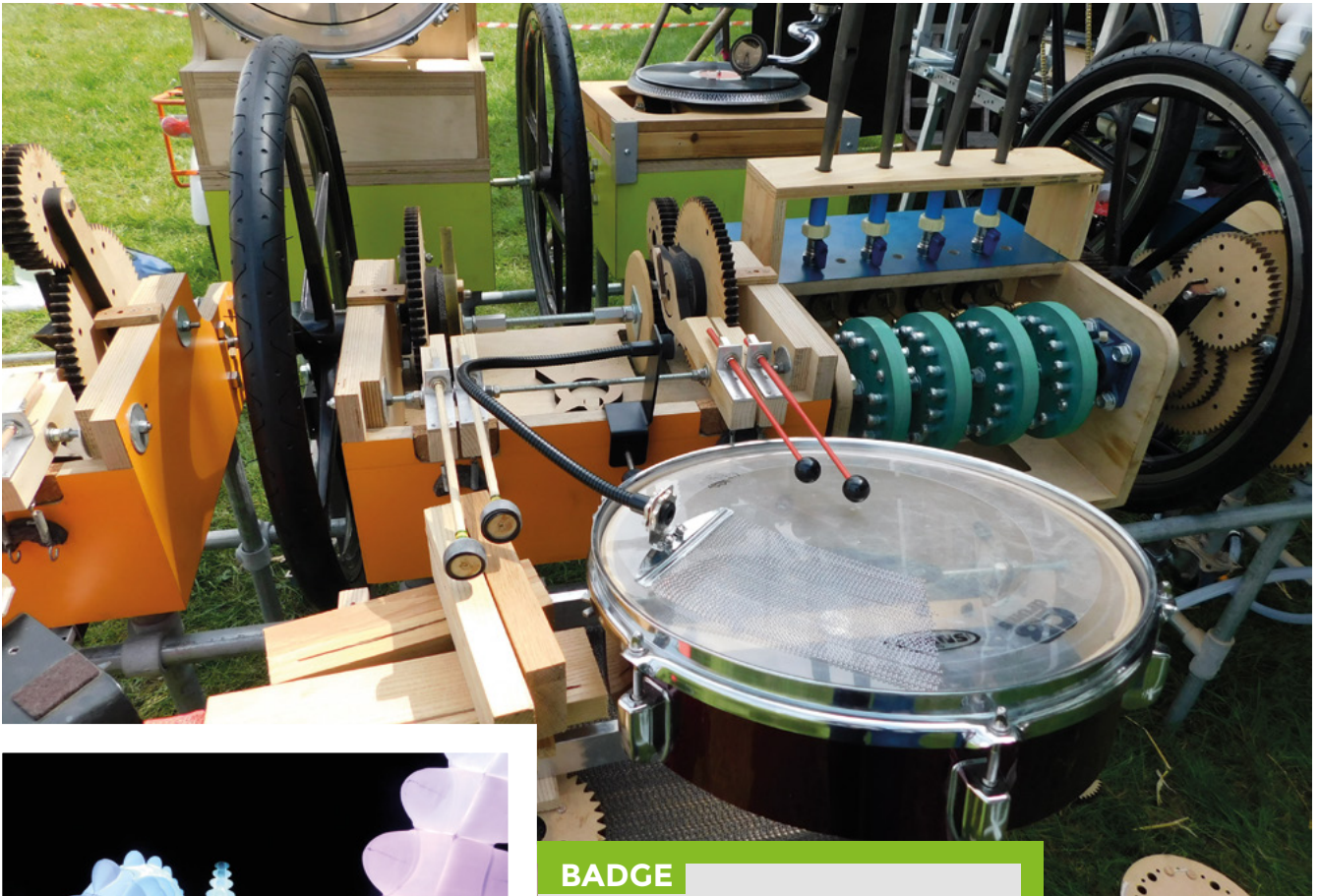
After being invited to join a fire art group at Burning Man, Jeff Gough decided to build a flaming top hat. Jeff talked us through the process of designing and making something that's small enough to fit on your head, but also safe and fun. He wore the result – a hat that emitted occasional puffs of burning gas throughout the talk.

Being YouTubers

James Bruton and Matt Denton (Ruth Amos was meant to join in as well, but had to cancel at the last minute due to Covid) talked us through their experiences building YouTube audiences, and bonding over their different builds of BB-8.

Rewilding human-computer interaction

Artist Tim Murray-Browne thinks that the way we work with computers has become too sanitised. While he didn't provide an ultimate solution, he's been working with dancers to create a movement-based interface. In this interface, machine learning (ML) translated movements into sound in a non-



BADGE

Every attendee got a programmable electronic badge. Known as 'TIDAL', this featured an ESP32 processor (with WiFi and Bluetooth), a small TFT screen, joystick, two buttons, three-axis accelerometer, three-axis magnetometer, and a battery.

You can create and install MicroPython apps from the App Store. You can see the available software at 2022.badge.emfcamp.org. Options include a very basic oscilloscope, some games, and a weather forecast.

The choice of MicroPython and an online editor meant you didn't have to install any software to create your own apps. Just plug in your badge (which featured a USB-C socket that slotted straight into a USB port on a computer – no cable needed), point your browser to editor.badge.emfcamp.org, and you were ready to start coding.

This toolchain-less setup made it much easier to start creating things while on site.

Above ♦
The Mammoth Beat Organ mechanical instrument in action

linear way. A simplistic movement (such as waving an arm) doesn't in itself result in a particular sound. Instead, the computer tracks a range of movements and converts them into sound using a dancer-specific ML model.

Alongside these talks were workshops where people could learn to solder, build, sew, blacksmith, and other maker skills. The workshops took place in 'villages' – these are self-organising groups of attendees. Anyone can start a village where they and others can camp together. Some villages included larger tents and gazebos which were used as meeting places and general areas to relax and chat.

Standing apart from the rest of the main festival was Null Sector. At the end of a track that leads you

through the camping field stood a tower of button-activated flaming torches, and more lasers than had any right to be in a field, lighting up the sky across the whole event. Inside was a curious mash-up of nightclub, market, escape room, and interactive art display. Oh, and there was a scanning electron microscope as well. It's the sort of place that defies easy categorisation. →



Right 
We're slowly
building up the
skills to make our
own Hacky Racer



BY GEEKS FOR GEEKS

EMF Camp is unashamedly a festival run by geeks – the organisers and setup crew are all volunteers. Nowhere is this more obvious than in the infrastructure. There's power available to your tent, and WiFi coverage across the entire site. This alone would be quite impressive, but it doesn't stop there. There's also a DECT phone network that allows you to use a regular wireless phone – not a mobile, but the handset of a landline. There's an analogue phone network that lets you use regular landlines, as well as fax machines and dial-up modems. It's not just old-fashioned tech, though. There was also an experimental 4G cell.

If you're wondering what use a DECT network, or a fax-compatible analogue phone network is, then you're missing the point entirely. None of this is essential, and certainly not in a grassy field in Herefordshire. There is no need to have dial-up internet when WiFi is everywhere. But then WiFi doesn't go *gruuuhhhh, hurrrrrrrr, badoing, badoing, kruuurr* when you connect, and that noise is, for a certain generation of geek, the sound track of our childhood. EMF Camp is a biannual chance to dust out the old computing equipment, don some rose-tinted glasses, and remember how things used to be.



Speaking of old equipment, there was an arcade machine full of games of yesteryear. A personal highlight was a 1979 version of Asteroids running on a vector display. Vector displays mostly came and went before I reached the age of arcades, so this was the first one I'd been able to play. While I've played many raster re-creations of this game, I was genuinely blown away by the vector graphics in a way I wasn't expecting. I think that perhaps my eyes have come to expect and ignore slight pixelation, so when greeted with a truly straight line from a cathode ray, my brain struggled to process the reality.



RACE-READY

One of this author's personal highlights this year was Hacky Racers. This is where people (or teams of people) build small novelty electric go-carts and race them around a field. It's broadly based on the Power Racing Series in the USA. Points are awarded for style as well as speed on the track, so designs included a giant Meccano vehicle, a cut-and-shut Little Tikes car, and an *Only Fools and Horses* yellow three-wheeler that appeared to be stuck in reverse.



I was invited by race organiser Mark Mellors to have a go in the 'Rule Zero' fire engine. The racing vehicles have a maximum width of 90cm and length of 150cm, which results in quite a snug fit. After wriggling in, I set about a few laps of the empty track. It was great fun. In the two weeks that I've been back home since EMF Camp, I've already bought two broken 'hoverboards' to salvage motors from in order to build my own. Look out for more on the build in future issues of HackSpace magazine.

It's difficult to sum up an event like EMF Camp in a few pages, but hopefully we've given you a flavour of what it was like. If you'll be in the UK in 2024, and are interested in technology or making things, we highly recommend you go. ■

THANKS!

EMF Camp is the product of hundreds of volunteers. The people who give the talks, the people who run the bar and shop, and even the event organisers are volunteering. It's the passion that these people bring that makes the event special.

We'd like to give a heartfelt thanks to all who helped make this event great.

If you're thinking of attending in 2024, please consider volunteering, whether that's by giving a talk, running a workshop, or doing a shift in the bar.

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction), let us know at hsmag.cc/hello

SUSTAINABLE MAKING

Alex Glow makes some very good points in your interview with her last issue. Whenever I make anything, I must go through a few iterations before I get anything to work – that means multiple PCBs made (with multiple boards shipped from the factory), multiple loads of solder, components, and, even when they do work, everything needs a battery, with all the pollution their production entails, and that battery needs electricity. That all sounds a bit grim, but one thing Alex said stands out: "It's not our fault, but it is our responsibility". We can try to do better without feeling guilty about the 60–70 years of oil extraction and plastics waste, because it's overwhelmingly the fault of a handful of companies. All we can do is try to do better, and not let the perfect be the enemy of the good.

James

Sunderland

Ben says: Alex has put together a great resource for makers who want to minimise their contribution to the forthcoming environmental apocalypse – green-ee.com. We've found it was worth a browse, even if you don't have a project in mind, as it's helped steer our thinking without our even realising it. It's easy to feel helpless, but small changes gather momentum.





PICO W

My MagPi subscription just dropped through my door – and there's a microcontroller attached to it! Now I just need someone to come up with a load of projects for me to copy. Can anyone out there help me?

Lucy

Bedfordshire

Ben says: It was our pleasure to give away the Raspberry Pi Pico to our subscribers when it launched in January 2021; now there's another version of this affordable microcontroller board, but with added internet connectivity. And yes – as you'll have discerned by now, there's a heap of ideas for you to make with your new Pico W, starting on page 32. Have fun!

ROCKET GLIDER

The swing-arm rocket glider that your writer made last issue was brilliant, and gave a perfect example of how the judicious application of 3D-printed parts to a more traditional build can be so much better than trying to 3D-print the whole thing. Dare I ask if we can hope to see more rocketry projects in the near future?

David

Germany

Ben says: As luck would have it, our Rocketry Correspondent, Jo Hinchliffe, is right now working on a flat-pack rocket as a tutorial for the magazine – readers are invited to think of a fake Swedish name for it, were it ever to arrive on the shelves of IKEA.



CROWDFUNDING NOW

ShaRPiKeebo

The little computer with big battery life

From \$150 | hsmag.cc/Sharp | Delivery: November 2022

Building a portable computer is easier than it's ever been. With tiny single-board computers such as Raspberry Pi, you just need to find a small keyboard and screen in the form factor you want, strap on a battery, and you're ready to go.

However, there's a big difference between building a computer and building a good computer, and that comes down to understanding your needs and picking exactly the right components for the job.

The ShaRPiKeebo is small and packs a long battery life. The former is thanks to the Raspberry Pi Zero W (or Zero 2 W), and the latter is thanks to the SHARP Memory Display. This display is almost like a cross between e-paper and an LCD screen. It takes very little battery, and is just a single colour (like e-paper), but has much faster update times (like an LCD screen). For on-the-go computing – particularly if

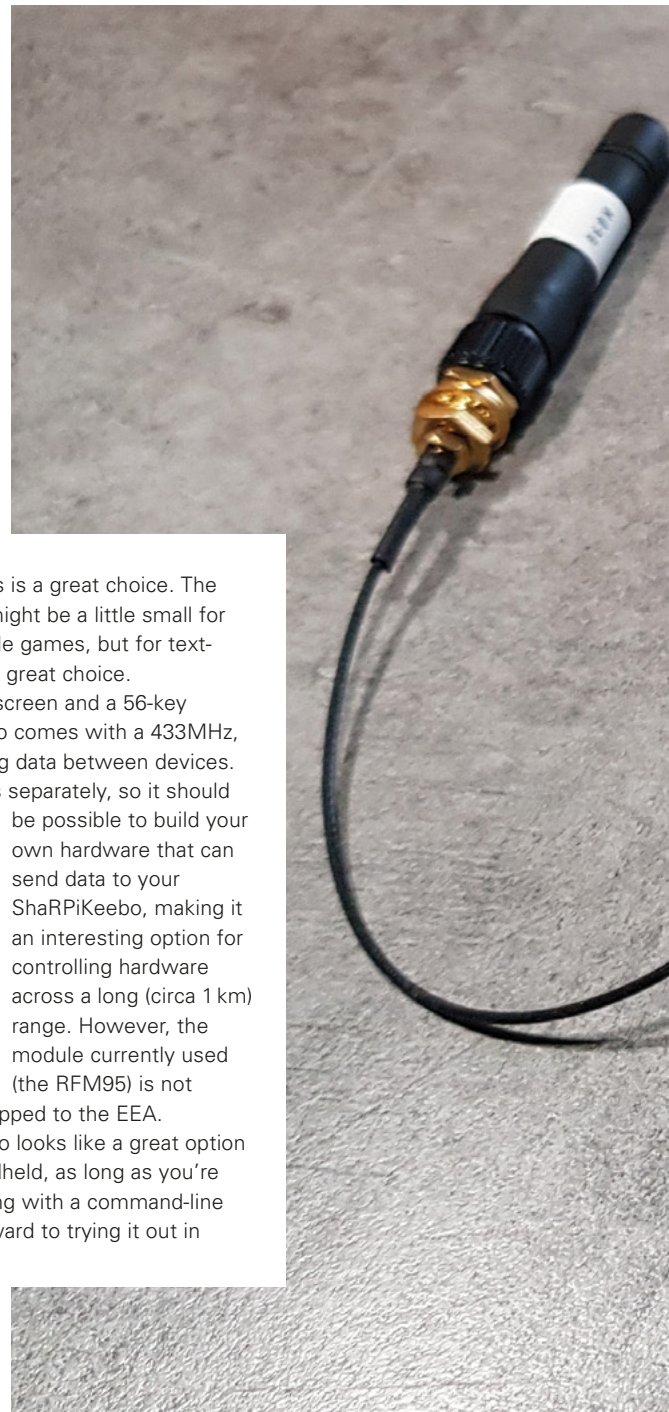
it's mostly text-based – this is a great choice. The 400×240 pixel resolution might be a little small for graphical use beyond simple games, but for text-based things, it looks like a great choice.

As well as a low-power screen and a 56-key keyboard, the ShaRPiKeebo comes with a 433MHz, long-range radio for sending data between devices.

You can get these modules separately, so it should be possible to build your own hardware that can send data to your ShaRPiKeebo, making it an interesting option for controlling hardware across a long (circa 1 km) range. However, the module currently used (the RFM95) is not CE-marked, so can't be shipped to the EEA.

Overall, the ShaRPiKeebo looks like a great option for a battery-powered handheld, as long as you're mostly interested in working with a command-line interface, and we look forward to trying it out in real life. □

The ShaRPiKeebo looks like a great option for a battery-powered handheld



BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.

Left

The SharRPiKeebo is small enough to fit in one hand



enviro goes wireless

Flexible Power
Pico W Aboard
Long Battery Life
Human Friendly
Real Time Clock

Alkaline NIMH Micro-USB LIPO

Wireless, standalone boards

Run for weeks or months on batteries

Working examples and custom libraries

Efficient sleep and scheduled readings



Automation 2040

RELAY AND IO CONTROL UP TO 40V WIRELESSLY
FOR HOME, OFFICE AND INDUSTRIAL AUTOMATION

PICO W
ABOARD



<https://pimoroni.com/auto2040w>

INKY FRAME 5.7

Your seven colour E Ink pal who's fun to be with



<https://pimoroni.com/inkyframe>

PICO W
ABOARD

PIMORONI

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG
46



HOW I MADE: BEAM ROBOT

Analogue, solar-powered
robotics – and it looks as cool
as it sounds

PG
52

INTERVIEW: ANDREW SINK

3D printing, photogrammetry,
and why you should print all
the Baby Yodas you like



PG
60

IMPROVISER'S TOOLBOX: BARBED WIRE

It's strong and spiky, but it's also
bendy – why barbed wire should
be in your toolbox

PG
32

PICOW

PROJECTS

Add wireless power to your
Raspberry Pi Pico projects

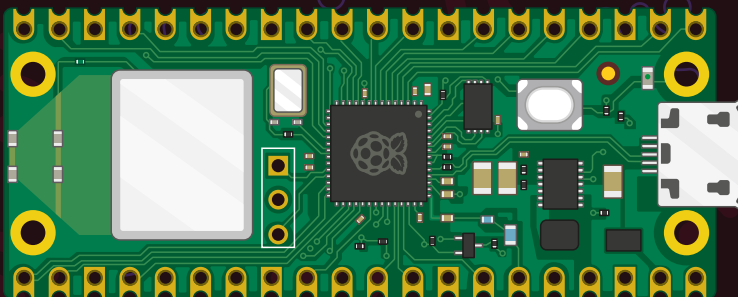
PG
64

IN THE WORKSHOP

Summer is here, so
it's time to cook meat
over a hot fire

PICO W

PROJECTS





he Pico W is, rather predictably, a Raspberry Pi Pico but with added wireless networking. For just

\$6, you get a dual-core Arm Cortex M0+ microcontroller

running at up to 133MHz with 264kB of RAM and 2MB of flash storage, and on top of this is an Infineon CYW43439 which adds wireless networking (the chip also supports Bluetooth Classic and Low Energy, but at present there's no firmware support for this).

All the things we've come to know and love from the original Pico are still there. There's still a huge bundle of input and output options, including Programmable I/O. In fact, the only slight change is that the on-board LED is now attached to the wireless chip rather than the main microcontroller (see box on page 35).

Not only does it have all the old features, but Pico W is in the same form factor as Pico, so adding networking to a project is usually just a case of swapping a Pico for a Pico W.

In this article, we're going to take a look at some great projects to help you get started and get the most out of Pico W. ➡

JOIN THE ONLINE COLOUR-CHANGING

LED COMMUNITY

Add some light to your electronics

While Pico W is very similar to Pico, you will need to get the special wireless version of the MicroPython firmware. You can download this from hsmag.cc/DocMicroPyth.

Make sure that you select the version for Pico W, as the version for the regular Pico won't work. You should have a UF2 file, and this can be loaded onto Pico W.

First, make sure that Pico W is unplugged from USB, then hold down the BOOTSEL button and plug Pico W into your computer using a micro USB to USB cable, then release the button. You should notice a new USB drive appear on your computer. You can drag and drop the UF2 firmware file onto this drive. Once it's copied

over, the drive will disappear and Pico W will reboot into MicroPython.

We recommend Thonny for working with MicroPython (though you can connect over serial as well). If you haven't already got this, you can download it from thonny.org – it works on Windows, Mac, and Linux (including Raspberry Pi). Open up Thonny and it should automatically detect and connect to MicroPython running on Pico W. In the bottom box, you should see a line something like this confirming it:

MicroPython v1.19.1-88-g99c258977 on 2022-06-30;
Raspberry Pi Pico W with RP2040

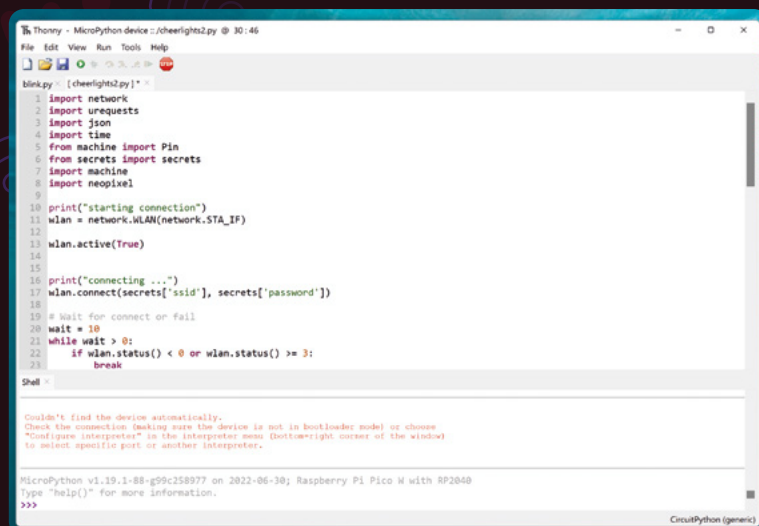
If you see that, then you're all set up and ready to go.

For our first internet-connected project, we're going to create a CheerLight. The idea of this is to have a colour-changing LED linked into a global system of setting colours. You can set the colour by tweeting [@cheerlights](https://twitter.com/cheerlights) and including a colour name in the tweet. This will change LEDs across the world to the colour you name. In an increasingly divided world, having an LED change colour is just a little reminder of how we're all connected. This can sit in the corner of your office and remind you, in a small way, that you're not alone. You're part of a worldwide community of makers.

We'll be using WS2812B LEDs (also known as NeoPixels) as our light. These can be programmed to display a wide range of colours. These LEDs come in a few different forms, but most commonly strips. You can use any form you like.

First, let's wire it up. Your LEDs should have three inputs, usually in the order 5V, Data In, and GND. 5V and GND should be connected to VBUS and GND on Pico W,

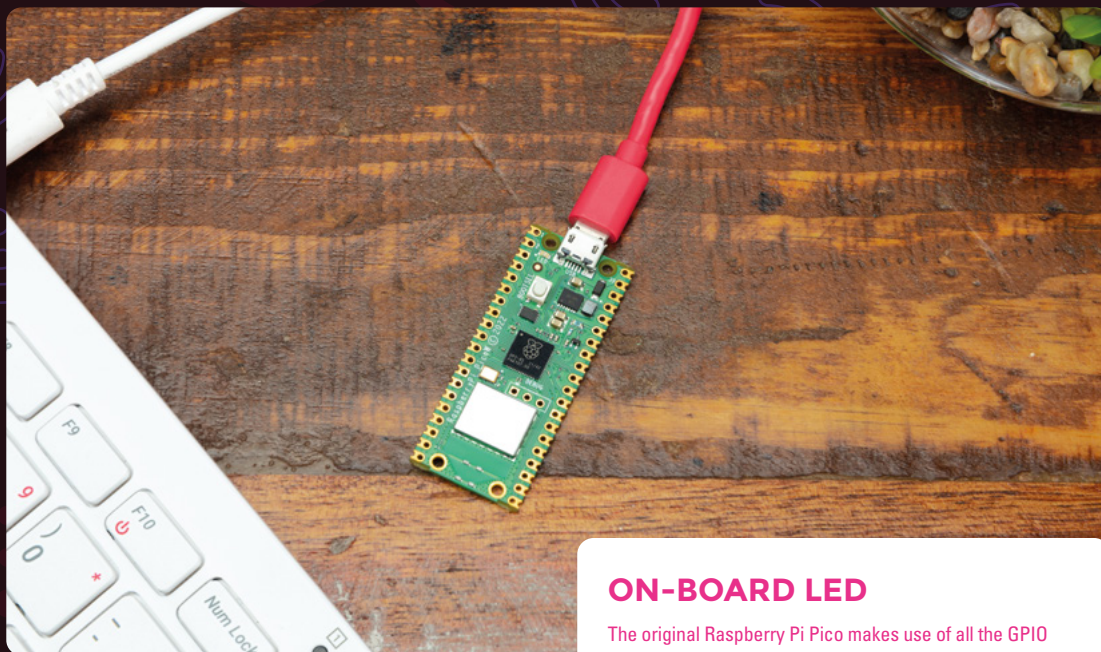
Below  The Thonny MicroPython IDE makes it easy to program and transfer files to your Pico



```

1 import network
2 import urequests
3 import json
4 import time
5 from machine import Pin
6 from secrets import secrets
7 import machine
8 import neopixel
9
10 print("starting connection")
11 wlan = network.WLAN(network.STA_IF)
12
13 wlan.active(True)
14
15
16 print("connecting...")
17 wlan.connect(secrets['ssid'], secrets['password'])
18
19 # Wait for connect or fail
20 wait = 10
21 while wait > 0:
22     if wlan.status() < 0 or wlan.status() >= 3:
23         break
24
25
26 
```

MicroPython v1.19.1-88-g99c258977 on 2022-06-30; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>



Left ■ You can program Pico W from most computers, including a Raspberry Pi 400

while Data In should be on GPIO 18 (you can use another GPIO pin if you like, but you'll need to change it in the code). You can solder these on or connect them with jumpers and crocodile clips.

CODING IT UP

We need to connect Pico W to the wireless network. Obviously, this needs an SSID (network name) and password. You can put these directly in your MicroPython code, but this can be a bit inconvenient because it means you have to be a bit careful about uploading your code to an online code repository, and it means you have to type the details into the code each time you take them from a repository onto your device. Instead, we're going to use a secrets file. In this, we'll put all the things that we don't want to be publicly known. It'll include the WiFi login details, as well as secret keys for things we'll be using in other examples.

```
secrets = {
    'ssid': 'XYZ',
    'password': 'XYZ',
    'aio_key': 'XYZ',
    'aio_username': 'XYZ',
    'ifttt_key': 'XYZ',
}
```

Enter that in Thonny and press Save, select 'MicroPython device', and call it **secrets.py**.

Now, let's connect to the network:

```
import network
import urequests
import json
import time
```

ON-BOARD LED

The original Raspberry Pi Pico makes use of all the GPIO pins. However, some are now needed to attach the wireless module. One solution was to reassign GPIO 25. On Pico, this is attached to the on-board LED, but on Pico W, it's attached to the wireless module.

The LED remains, but now it's controlled by the WiFi module rather than the main microcontroller. Fortunately, much of this is hidden by MicroPython, and there is a pseudo pin called 'LED'. You can blink the LED with:

```
import machine
import time
led = machine.Pin("LED", machine.Pin.OUT)
while True:
    led.toggle()
    time.sleep(0.5)
```

This pin should behave as any plain pin does, but it can't be used with PIO, PWM, or other special functions.

```
from machine import Pin
from secrets import secrets
import machine
import neopixel

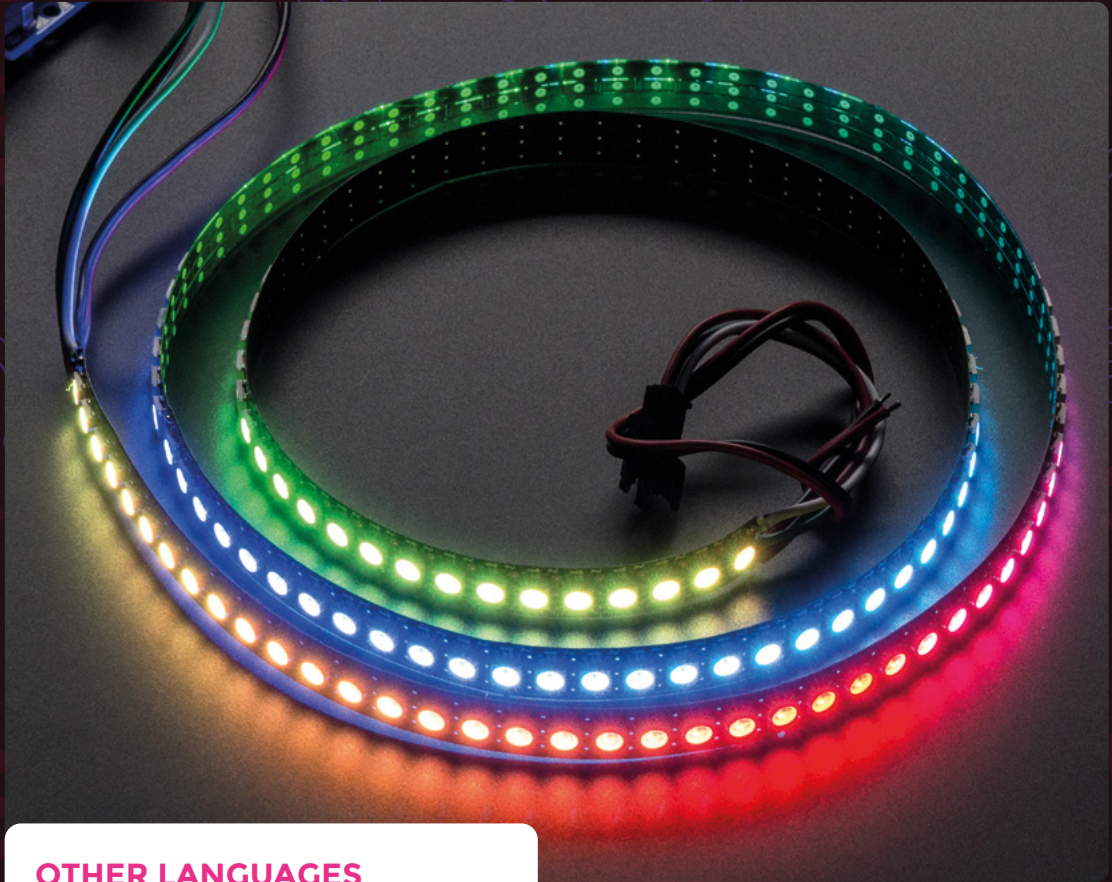
ssid = secrets['ssid']
password = secrets['password']

print("starting connection")
wlan = network.WLAN(network.STA_IF)

wlan.active(True)

print("connecting ...")
wlan.connect(ssid, password)

wait = 10
```

Right ♦ WS2812B LEDs come in all sorts of shapes and sizes, but they're all controlled in the same way

OTHER LANGUAGES

We love MicroPython because it lets us get projects up and running really quickly. However, there are times when another language is more appropriate.

C/C++ The official C/C++ SDK is available at: hsmag.cc/PicoSDK.

This supports networking through the LWIP networking stack and gives you the ability to work with the networking at a very low level. There are some examples to help you get started at hsmag.cc/PicoExamp.

At the time of writing, this is the only other language to support the wireless networking. However, we expect there to be other options very shortly.

Arduino Since the Arduino port of Pico includes the C/C++ SDK, it should be possible to use this with WiFi once the new SDK is brought in. However, at the time of writing, we were not able to test this.

CircuitPython Adafruit's flavour of Python for small devices typically gets new features quickly, so we would expect WiFi on Pico W to work very soon. It's likely to be based on the requests module, which is functionally very similar to urequests, which we use here.

Rust Pico has proved popular with the embedded version of this memory-safe language, and there is an active community working with this.

```
while wait > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    wait -= 1
    print('waiting for connection...')
    time.sleep(1)
```

```
# Handle connection error
if wlan.status() != 3:
    raise RuntimeError('wifi connection failed')
else:
    print('connected')
```

```
status = wlan.ifconfig()
print( 'ip = ' + status[0] )
```

We also need a bit to set up our LEDs:

```
pixels = neopixel.NeoPixel(machine.Pin(18), 10)
```

Now that we're online, we need a way of grabbing the colour. There's a back-end system that grabs colour changes from Twitter and posts them to the URL api.thingspeak.com/channels/1417/field/2/last.json in JSON format. We just need to download this and extract the 'field2' data. This is hex encoded, so we need to extract the red, green, and blue values from this and set our LED accordingly.

As long as we're connected to the internet, we can use the `urequests` module to do just this, and the `json` module will convert a JSON string to a Python dictionary. The following code grabs the latest colour from the URL and returns a colour:

```
def get_colour():
    url = "http://api.thingspeak.com/channels/1417/field/2/last.json"
    try:
        r = urequests.get(url)
        print(r.status_code)
        if r.status_code > 199 and r.status_code < 300:
            cheerlights = json.loads(r.content.
decode('utf-8'))
            print(cheerlights['field2'])
            red_str = '0x' + cheerlights['field2']
[1:3]
            green_str = '0x' + cheerlights['field2']
[3:5]
            blue_str = '0x' + cheerlights['field2']
[5:7]
            colour = (int(red_str), int(green_str),
int(blue_str))

            r.close()
            return colour
```

```
    else:
        return None
    except Exception as e:
        print(e)
        return None
    return colour
```

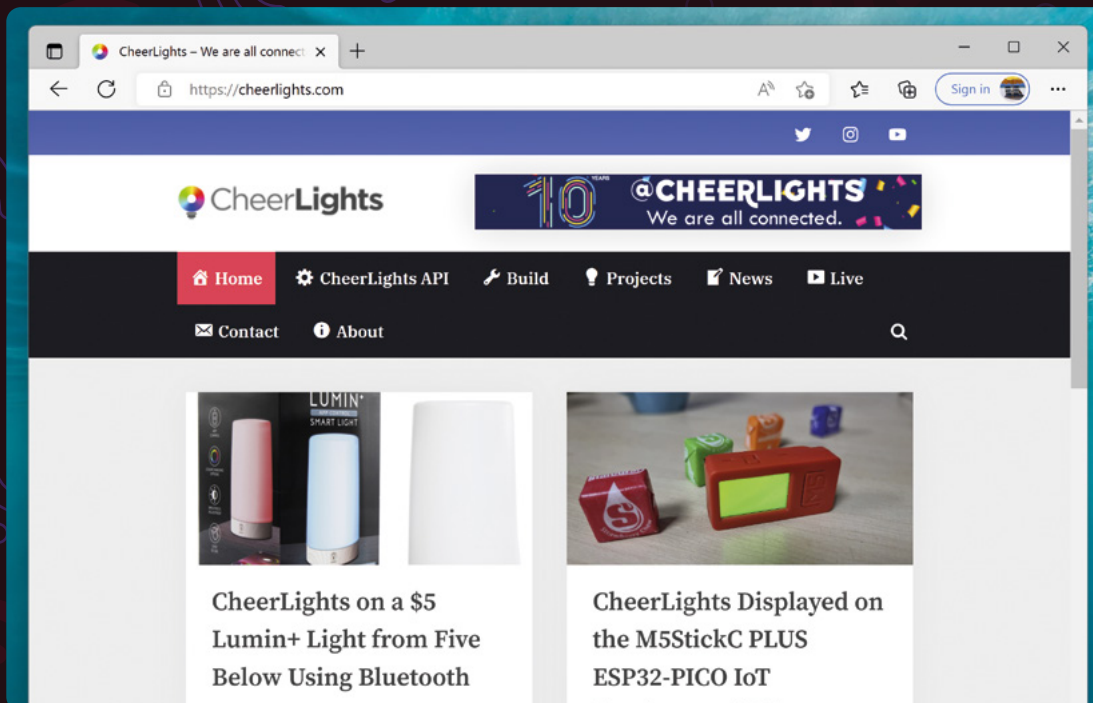
We have to do a bit of string manipulation to get the colours out of the string. First, we use slicing to get the two hexadecimal digits that relate to the colour, then we add '0x' to the start of these two digits. The '0x' lets Python know that the string contains hexadecimal characters, so we can then use `int()` to convert this new string into an integer.

Finally, we just need a loop that checks the colour every minute and updates the LED.

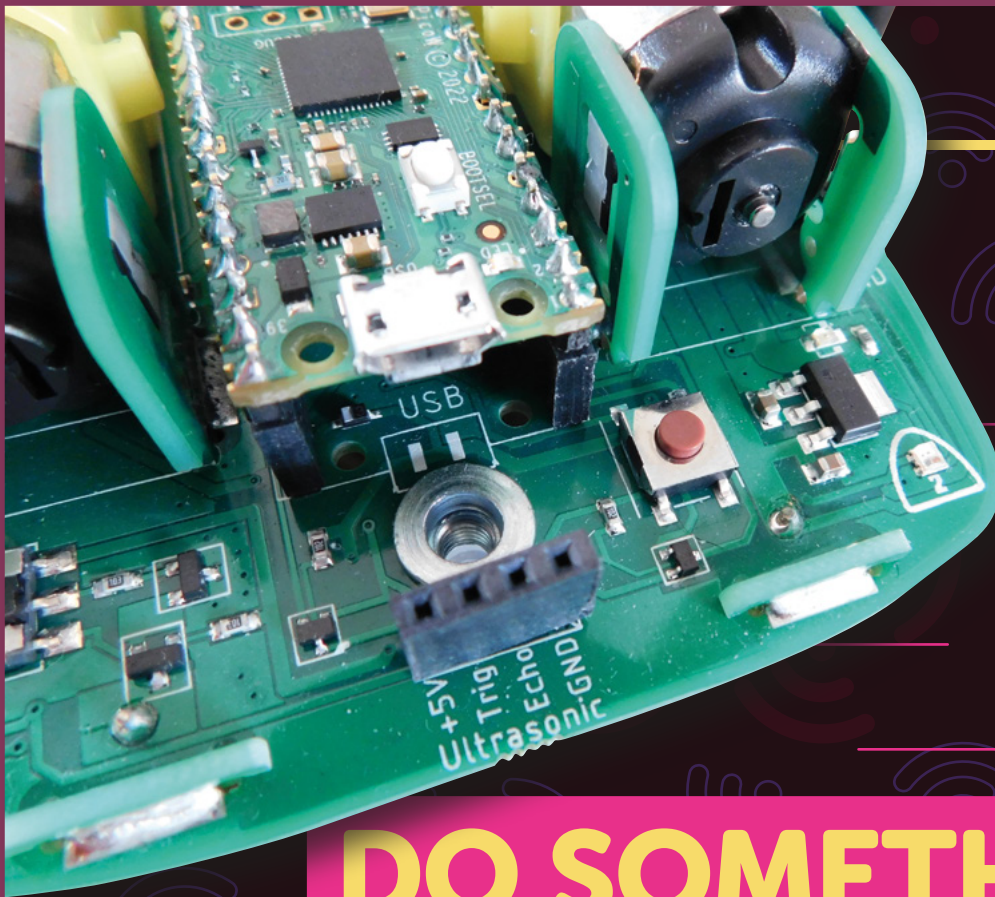
```
while True:
    print("getting colour")
    colour = get_colour()
    if colour is not None:
        pixels.fill(colour)
        pixels.write()
        time.sleep(60)
```

You can get the full code at hsmag.cc/cheerlight.

Load it up to your Pico and you should see the LED occasionally change colour. Alternatively, tweet to [@cheerlights](https://twitter.com/cheerlights) and send a message to lots of devices – including some Pico Ws – to change colour. →



Left ■
Find out more about
CheerLights at
cheerlights.com



DO SOMETHING

BUTTON

Use Pico W to control the cloud

Now that we've got data from the internet, let's send some to it and create a button that sends a message to the internet to do something. Do what, you ask? Well, do almost anything! The magic glue in this project is If This Then That (IFTTT), an online service that connects causes to events. This means that you can create a thing that triggers a huge range of possible events. In this example, we'll use it to control our heating using a Hive thermostat. However, IFTTT works with a huge range of different web-connected devices, so you can easily modify this to work with any device that's supported. Take a look here for a full range of supported systems: ifttt.com/explore/services.

If This Then That lets you create very simple programs without having to program anything. The programs are known as 'applets' and all need to have the form 'If this event happens, then trigger this action'. There's support for lots of different possible events and actions. The particular event that we'll be using is known as 'Maker Webhooks'. These create a URL that, if you visit it, will trigger an 'If This'.

The first part of this is to create an applet in IFTTT which we will trigger from Pico. Head to ifttt.com, create a free account (if you don't already have one), and sign in. You can then press Create to create a new applet. The first part to configure is the If This part. Press Add and search for webhooks. Inside this option, select Receive a Web Request, then give it an event name (you'll need to enter this in the program later on). Press Create Trigger to finish the first part.

Second, we create the Then That part. Again, click on Add. Here, you can configure your applet to do whatever you want. We've configured ours to send us an email, but you might want to browse through the list of options to see if there's some particular service you'd like to trigger.

Now, let's take a look at the Pico W side of things.

The only wiring is to connect a button. One side of this should go to 3V and the other side to GPIO pin 0.

We need to update our secrets file with a couple of bits of information for IFTTT: the event name and key. Technically, the event name doesn't have to be secret, but the secrets file is a good place to put it.

Above ♦
This code also works with the user button on the Kitronik Autonomous Robotics Platform for Pico

Add the following inside the **secrets** dictionary in the secrets file:

```
'ifttt_key': 'XXXX',
'ifttt_eventname': 'web_button',
```

You can find out your key by going to the URL **ifttt.com/maker_webhooks/settings**. There, you should see a section like:

URL

<https://maker.ifttt.com/use/<your key here>>

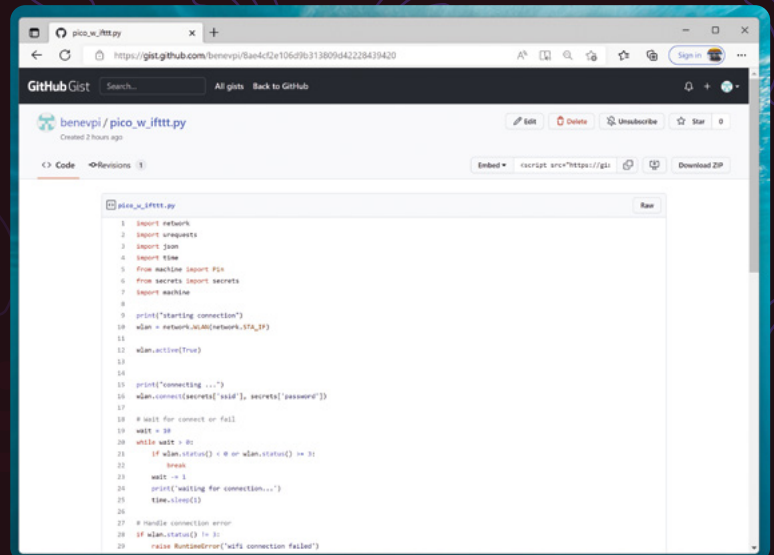
Where we've written **<your key here>**, you'll find your key and you can copy and paste that into the secrets file.

You can see the full code at **hsmag.cc/picow_ifttt**.

To start, it's exactly the same as the previous example, as we just need to connect to the internet.

Webhooks are triggered by visiting a particular URL. You can also attach up to three bits of data to your webhook. We can do this by adding **"?value1=<data>&value2=<data>&value3=<data>"** to the end of the URL. We won't actually use this in our example, but you may want it if you use this function in your own code.

```
def ifttt_webhook(event, key, values=None):
    url="https://maker.ifttt.com/trigger/"+event+"/with/key/"+key
    if values is not None:
        url = url + '?'
        for counter, value in enumerate(values, start=1):
            if counter < 4:
                if counter > 1:
                    url=url+'&'
                    url=url+"value"+str(counter)+"="+values[counter-1]
                try:
                    r = urequests.get(url)
```



```
if r.status_code>199 and r.status_code<300:
    r.close()
    return 0
else:
    r.close()
    return -1
except:
    r.close()
    return -2
```

Above ■ You can get the code from **hsmag.cc/picow_ifttt**

Below ■ Take a look at the available services on IFTTT to see if there are any that you use already

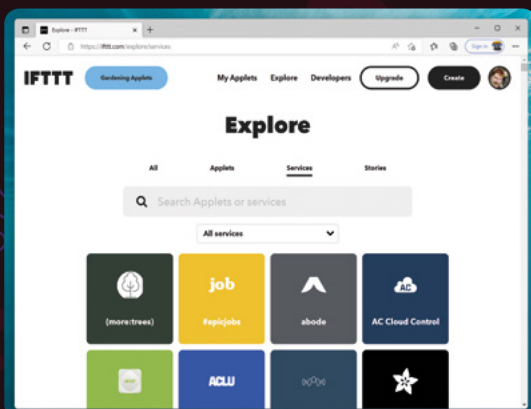
As you can see, this builds up a URL with the relevant values for event and key. You can also submit up to three values in a list, and these will also be passed to the webhook (IFTTT doesn't permit more than three). This uses the same urequests module that we used in the previous project.

Finally, we just need a bit of code to trigger this when a button is pressed.

```
button = machine.Pin(0, machine.Pin.IN, machine.Pin.PULL_DOWN)

current_state = 0
while True:
    new_state = button.value()
    if new_state != current_state:
        if new_state == 1:
            ifttt_webhook(secrets['ifttt_eventname'], secrets['ifttt_key'], values=None)
            current_state = new_state
            time.sleep(0.5)
```

This project is surprisingly powerful because of the range of integrations available from IFTTT. Exactly what you can use it for depends entirely on what online services you use. Potentially, you can use it to log particular events, control your heating, send alerts, even open doors and turn lights on. →



WEB SERVER

ROBOT

Control a buggy from the web

So far, we've looked at getting data from the internet and sending data to the internet, but in both cases, we connected to an external server. However, there are times where we don't want to send data to an external system, but use a wireless connection to send data from one computer to another.

In this case, you can go via the internet – for example, use one machine to send

data to Adafruit IO and another to read data from the same feed. Doing this is reliable and you can keep a trail of data, should you need to restart one of the machines. However, it's also slow, and there are times when you might not want to trust your data to the internet.

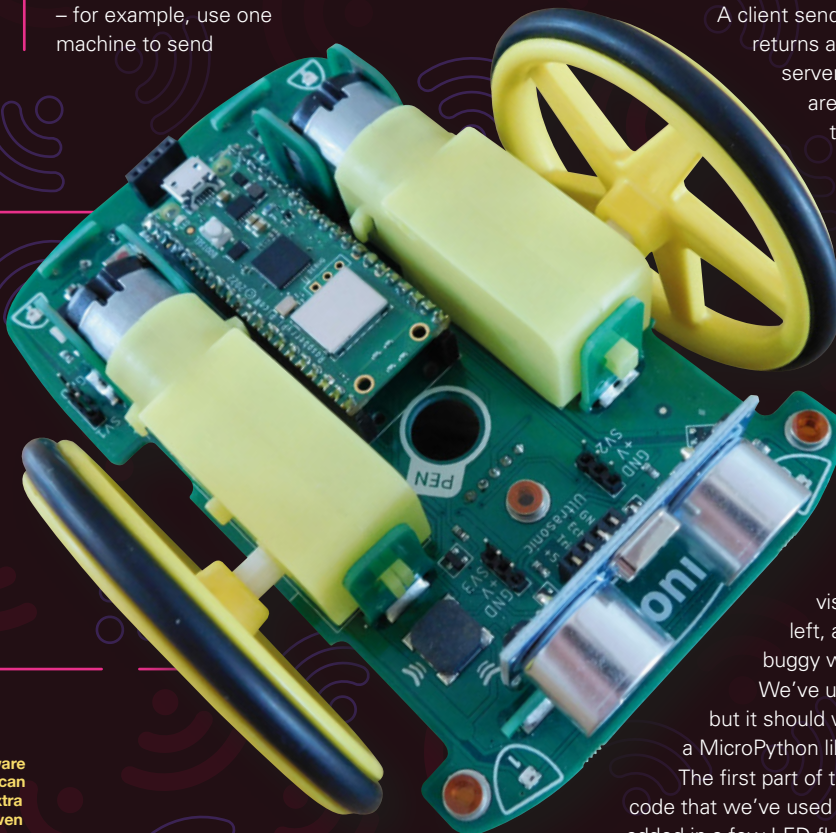
Fortunately, there is a solution. Hypertext Transport Protocol (HTTP) – the protocol that the web is built on – is actually a very simple text-based protocol.

A client sends a request to a server which returns a response. The sort of web servers that power most websites are complex bits of software, but this complexity comes from performance and the features that allow you to automatically place dancing baby GIFs on your web pages and suchlike.

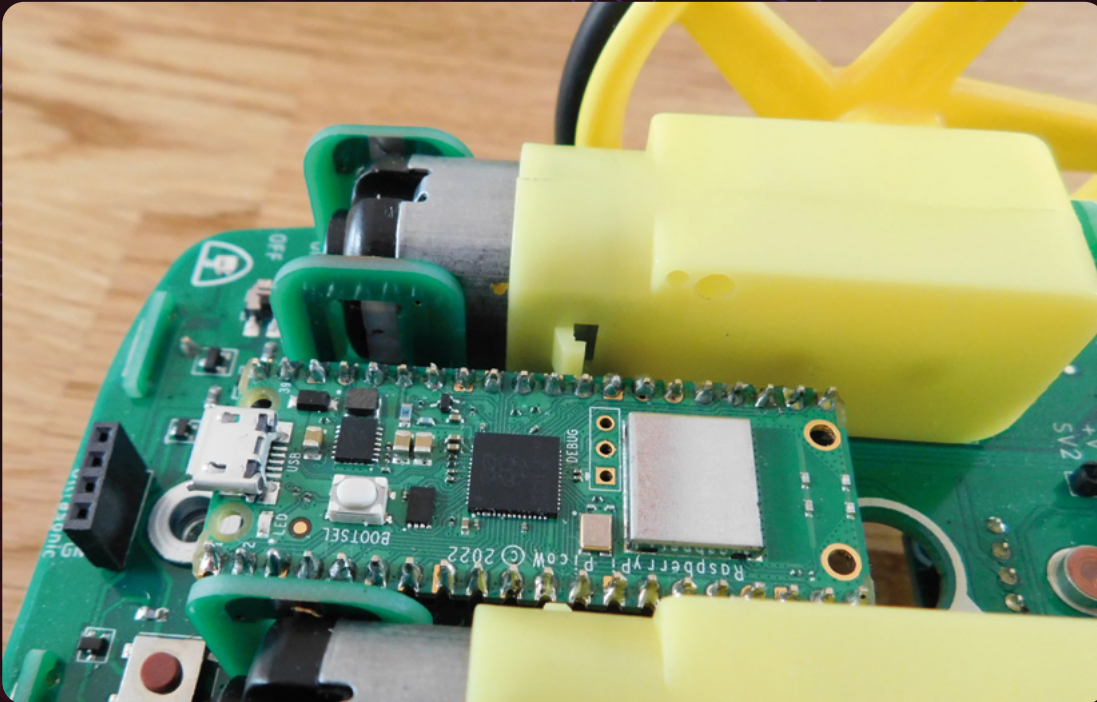
We can do away with all that and have a really simple web server that just controls a buggy. We want it to serve up the same web page whatever address the user puts in (which displays a simple remote control), and if they visit **/buggy/forward**, the buggy will move forward. If they visit **/buggy/left**, the buggy will go left, and if they visit **/buggy/right**, the buggy will go right.

We've used a Kitronik Pico Buggy for this, but it should work with any robot that you have a MicroPython library for.

The first part of the code is the same WiFi connect code that we've used in previous examples, but we've added in a few LED flashes on the buggy's four RGB



Right ♦ There's far more hardware than we've used – you can add extra servos, an extra distance sensor, and even a pen for drawing



Left ■ This robot was designed for Pico, but since Pico W is pin-compatible, it slots right in and works fine

LEDs so that we can see what's going on when we're not plugged into a computer.

```
import network
import socket
import time
from machine import Pin
from secrets import secrets
from PicoAutonomousRobotics import
KitronikPicoRobotBuggy

print("starting buggy")
buggy = KitronikPicoRobotBuggy()
print("started buggy")

buggy.setLED(0, buggy.YELLOW)
buggy.show()

ssid = secrets['ssid']
password = secrets['password']

print("starting connection")
wlan = network.WLAN(network.STA_IF)

buggy.setLED(1, buggy.YELLOW)
buggy.show()

wlan.active(True)

buggy.setLED(2, buggy.YELLOW)
buggy.show()
```

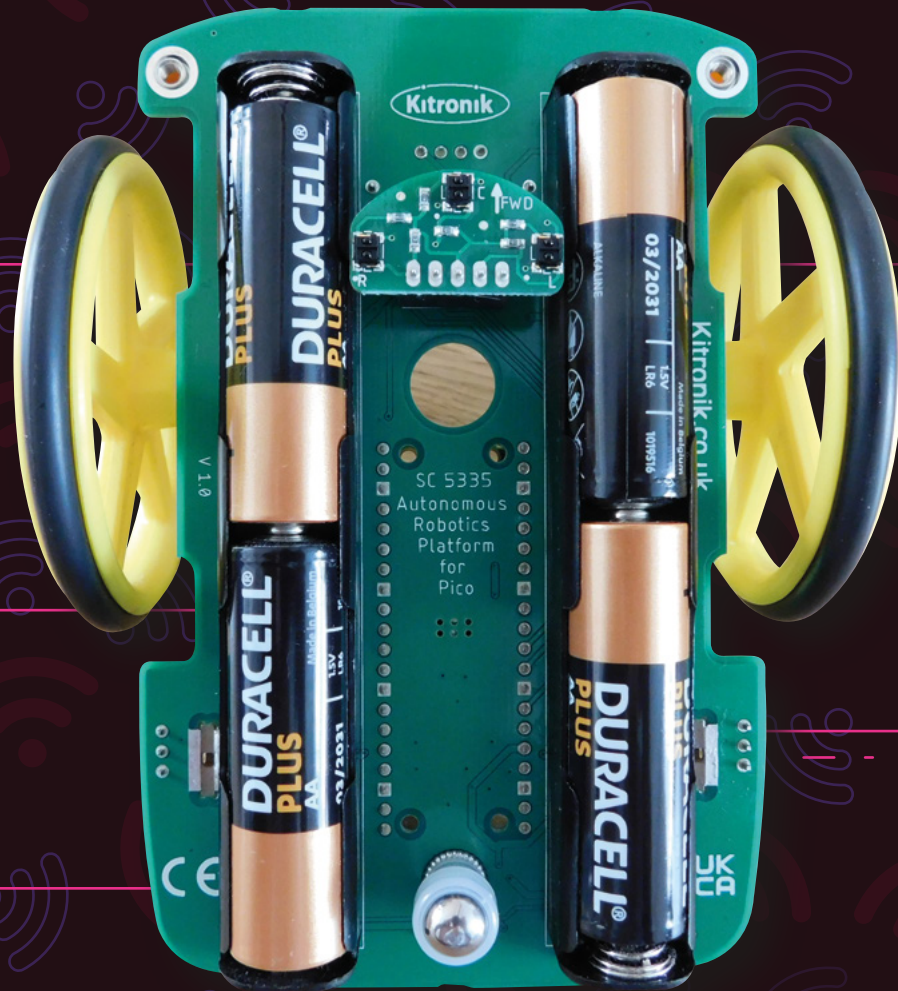
```
print("connecting ...")
wlan.connect(ssid, password)

buggy.setLED(3, buggy.YELLOW)
buggy.show()

# Wait for connect or fail

wait = 10
while wait > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    wait -= 1
    print(wait)
    if (wait % 2 == 0):
        print("here")
        buggy.setLED(0, buggy.BLUE)
    else:
        buggy.setLED(0, buggy.RED)
    buggy.show()
    print('waiting for connection...')
    time.sleep(1)

# Handle connection error
if wlan.status() != 3:
    raise RuntimeError('wifi connection failed')
else:
    print('connected')
    buggy.setLED(0, buggy.GREEN)
    buggy.setLED(1, buggy.GREEN)
    buggy.setLED(2, buggy.GREEN)
    buggy.setLED(3, buggy.GREEN)
```

Right ♦
Power comes from four AA batteries, which means this can run without any cables

```

    buggy.show()

    status = wlan.ifconfig()
    print( 'ip = ' + status[0] )
    # Open socket
    addr = socket.getaddrinfo('0.0.0.0', 80)[0][1]

```

Next, we need a variable that holds a string with the HTML response. We've kept this really short and simple, but it could easily be made more aesthetic, or include more information, if you want.

```

response = """<!DOCTYPE html>
<html>
<head> <title>Pico W</title> </head>
<body> <h1>Pico W</h1>
<p><a href="/buggy/forward">forward</a><br>
<a href="/buggy/left">left</a></br>
<a href="/buggy/right">right</a></p>
<h2> The next object in front is at %s</h2>
</body>
</html>
"""

```

We won't go into HTML in detail here as we are focusing on the Pico W, but links are between `<a href=link>` and `</a>`.

You might also notice the final line with **The next object in front is at %s**. Our buggy has an ultrasonic distance sensor that we can use. In Python, we can use `%s` to mark a point in a string where we want to include another string. We'll take a look at how to do that in a little bit.

HTTP runs on top of Transport Control Protocol (TCP). TCP has a range of numbered ports that we can bind to using the socket module.

Once we've created a socket object and bound it to a port and an address (since a device can in theory have multiple IP addresses, sockets are defined by both IP address and port), we can wait for an external device to make a TCP connection.

TCP connections themselves are really just a two-way text stream. You can use them to send and receive all sorts of data in much the same way that you can use the serial port. However, with HTTP, we use it in a very specific way. The client makes a request that is

text in a very specific format defining what page they want to access, as well as some details about the client and its capabilities. Again, we won't go into this in depth.

The important thing for us is that we take a look at the request, see if it's for the pages **/buggy/forward**, **/buggy/left**, or **/buggy/right**, and if it is, set the motors appropriately, then whatever page the user requested, returns the same HTML response defined above. The code for all this is:

```
def check_and_move(request, find_str, buggy, left_
motor, right_motor):
    request = str(request)
    if (request.find(find_str) == 6):
        buggy.motorOn("l", "f", left_motor)
        buggy.motorOn("r", "f", right_motor)
        time.sleep(1)
        buggy.motorOn("l", "f", 0)
        buggy.motorOn("r", "f", 0)

# Listen for connections
while True:
    try:
        cl, addr = s.accept()
        print('client connected from', addr)
        request = cl.recv(1024)
        print(request)

        request = str(request)
        check_and_move(request, '/buggy/forward',
buggy, 100, 100)
        check_and_move(request, '/buggy/left',
buggy, 100, 0)
        check_and_move(request, '/buggy/right',
buggy, 0, 100)
```

STARTING AUTOMATICALLY

You probably want your buggy to automatically start the program when you turn it on – it'd be a bit of a faff if, every time, you had to plug it into your computer and start Thonny just to launch the control program. Fortunately, it's really easy to do this in MicroPython – just save your code to the MicroPython device and call it **main.py**. If you do that, when you turn it on, it'll automatically start the code.

You still need to know what IP address the machine's on. Generally, routers will assign the same IP address to a device when it reattaches to the network, so you can run it once tethered and then try this again. If this fails, you'll need to log into your router to see what IP address is assigned – take a look at your router's documentation for details.

```
cl.send('HTTP/1.0 200 OK\r\nContent-type:
text/html\r\n\r\n')
cl.send(response % str(buggy.
getDistance("f")))
cl.close()

except OSError as e:
    cl.close()
    print('connection closed')
```

As you can see, we first have to accept a TCP connection, then we can receive data. Our **check_and_move** function takes the stringified version of the request as well as the page location we want to check and details of the motor settings. If it finds the **find_str** at position 6 in the request, then it knows that this is the page that the user requested. It turns the motors by the specified amount, waits for one second, then stops the motors.

OUR BUGGY HAS AN ULTRASONIC DISTANCE SENSOR THAT WE CAN USE

After this has happened, the main loop then sends the HTML to the client. Notice that it first sends a header line that has a bit of information about the data. This is required by HTTP. We can insert the distance into our HTML with the line:

```
cl.send(response % str(buggy.getDistance("f")))
```

The **%** command is used to substitute the **%s** in the original string with a new string – in this case a readout from the distance sensor.

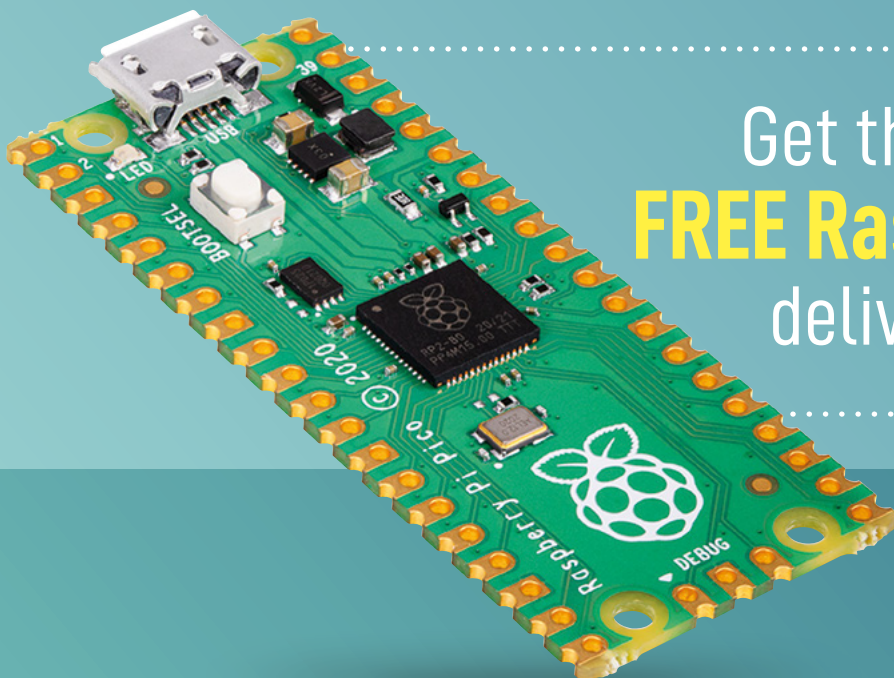
The only thing left to do is close the connection. You have to do this in order to free up memory – you'll quickly run out of memory if you don't.

Once this is loaded on your Pico and running, you can point your web browser to the IP address given in the console when you load the file (typically it'll stay the same between reboots, but it's not guaranteed to). Then you can click on forward, left, and right to move the buggy around.

Our little robot web app is obviously very simple, but you can take this technique and expand it in different ways to let you connect to and control all manner of different Pico W-powered devices. □

SUBSCRIBE TODAY

FOR JUST £10



Get three issues plus a
FREE Raspberry Pi Pico
delivered to your door

..... hsmag.cc/FreePico

UK offer only. Not in the UK?
Save money and get your
issue delivered straight to your
door at hsmag.cc/subscribe.
See page 66 for details.

Subscription will continue quarterly unless cancelled

SUBSCRIBE on app stores

From £2.29



Buy now: hsmag.cc/subscribe

Free Pico with print subscription only



MAKE | BUILD | HACK | CREATE
HackSpace

MAKE | BUILD | HACK | CREATE
TECHNOLOGY IN YOUR HANDS hsmag.cc | August 2022 | Issue #57

HackSpace

Robot Roundup RASPBERRY PI **PICO W PROJECTS**

Beer BBQ Turn a keg into an outdoor grill

Time-lapse Photography Watch life on fast forward

Make internet-connected builds with the **NEW \$6** microcontroller board

BEAM MACHINE ROBOTICS POWERED BY THE SUN'S RAYS

LASER CUTTING KNITTING WELDING DOGS

SAVE 44%

Aug. 2022 Issue #57 £6

PLUS Turn your Raspberry Pi into a production line

- Rocket Glider Like the space shuttle, but smaller
- CircuitPython PIO Sending data quickly

3D PRINTING POLYHEDRA

HACK | CREATE
HackSpace

ND S hsmag.cc | June 2022 | Issue #55

OLD A LOTTER

robotic machine with Raspberry Pi Pico

Jason Coon The Evil Genius of Fibonacci LEDs

Jun. 2022 Issue #55 £6

3D PRINTING / ARDUINO

HOW

By Jack Spiggle

MADE

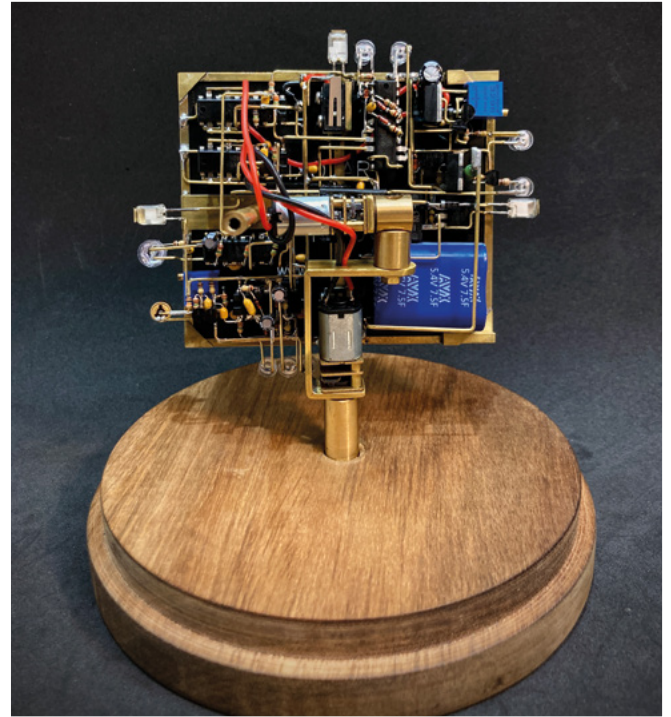
The term 'BEAM robotics' is one that would have been familiar to many makers in the early 2000s, but has faded into obscurity in recent years

with many of the old BEAM blogs and forums lost to time. 'BEAM' does not necessarily describe a specific form or function of any robot, but rather an ethos which is described in part by the name. The most widely recognised meaning of the acronym BEAM is Biology, Electronics, Aesthetics, and Mechanics. To me, two further elements define BEAM electronic design: the use of recycled components (which I personally no longer do these days, as new components are so cheap and readily available), and the simplicity of the circuits. These simple circuits are the most striking difference between BEAM and modern robots as BEAM robots lack any sort of microcontroller. That's right, we're talking analogue robots!

BEAM robots utilise a lot of passive components and other fundamental components like CMOS inverters, but lack any kind of microcontroller. The most complex BEAM robots are made up of blocks that mimic neurons. That is, once a certain stimulus threshold is met on the input of each block, an action occurs.

A SOLAR-POWERED ROBOT FROM OLD FASHIONED PARTS

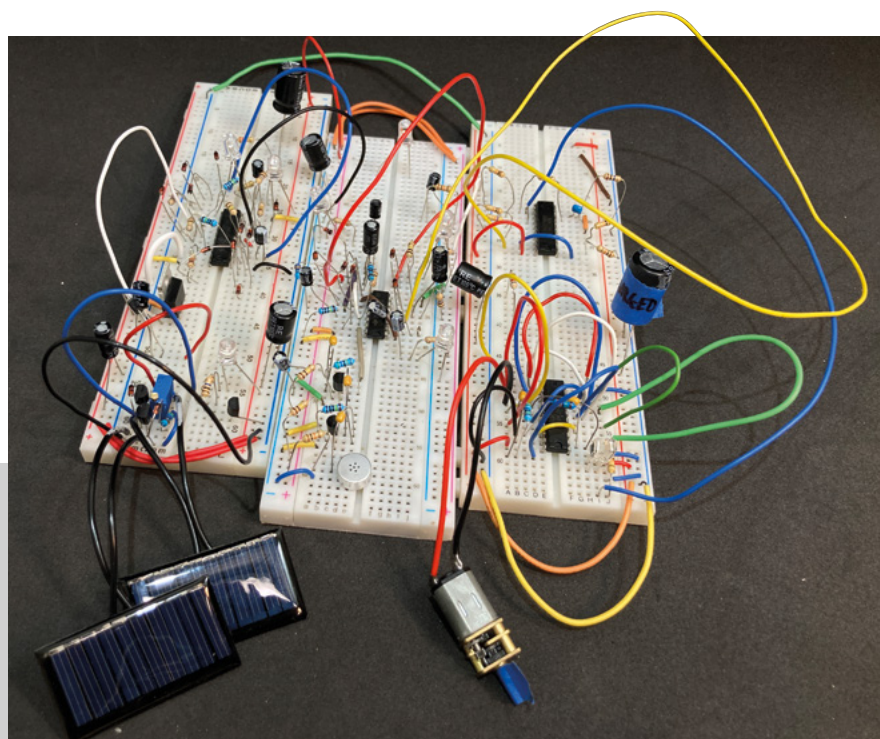
A design inspired by biology



Valentino Braitenberg's early thought experiments into artificial intelligence (and I mean 1980s early) showed how complex behaviours can arise from a robot created with just a few of these artificial neurons. Such neurons can easily be simulated on a microcontroller, yes, but they can be made even more simply with just a CMOS inverter. This also means that the input and threshold voltages remain entirely analogue. Perhaps this is why I still find BEAM circuits so endearing today: the entire system is built from the ground up, there is no abstraction of signals with 1s and 0s, and no abstracting of logic with high-level code. It really feels like you are manipulating electrons at the lowest level in order to create an intelligent, almost life-like system. I think that all of this is secondary to my robot's main function, which is to look pretty. I definitely consider my BEAM circuits to be more sculpture than robot, and few BEAM roboticists (bar Mark Tilden, the father of BEAM) ever created such complicated robots.

THE ROBOT

Enough with the philosophy, let's talk about the robot. What does it actually do? It sits in a glass bell-jar from IKEA, charging via the large solar panel and turning to face ➤



**“I ABSOLUTELY
ADORE WORKING
WITH BRASS”**

the brightest light source. It spends most of the day sitting rather stationary, only to move about quickly, like an excited dog, when in direct sunlight or when you pass your hand over it. Sitting on my desk, it generally wakes up around 8–10:00 in the morning and stops anywhere from

5–10:00 in the afternoon. To make it more interactive, I also added a circuit that makes the robot do a little dance when you give it three whistles.

MECHANICS

The mechanical assembly of this robot is centred around the motors and the solar panel. The motors are common N20 gear motors which were chosen for two reasons: 1) if the robot is still operating decades from now and a motor breaks, there will likely be replacements available, and 2) a wide variety of gear ratios were available to experiment with. In hindsight, I wish I had chosen a higher-quality motor as the N20 motors are noisy and have backlash that made the robot very hard to tune. The motor mounts and other brass components were all made by hand using tools like a hacksaw, drill, soldering torch,

and my favourite two tools, a jeweller's saw and needle files. I absolutely adore working with brass: it is easy to cut/bend, readily available in all shapes and sizes, solders well, and looks beautiful!

The solar panel is housed in a frame made from square brass tubing (most of the brass is from K&S Precision Metals). This brass frame was very important as everything builds off it both mechanically and electrically. Because I didn't want to damage the solar panel while building, I made the frame such that one side can be removed for the panel to slide out. Wherever possible, I tried to make the robot dismantlable in favour of permanent connections, in the hopes that the robot will still be serviceable decades from now.

ELECTRONICS

Onto the electronics, and the circuitry can be broken down into three major sections: the solar charging, the motor control, and the whistle/dance circuit. A lot of the robot was based on BEAM circuits originally designed by Wilf Rigter in the early 2000s with a lot of modifications and adjustments. The full circuit is much too big and nuanced to fully cover here, but there is an Instructable and a half-hour YouTube video overviewing the circuit in more detail.

The circuit is entirely free-formed, meaning that there is no PCB. Connections between components are created in three-dimensional space using bent brass wire, which I think looks stunning and creates a kind of electronic, steampunk vibe (BEAMPunk?). The process of free-forming feels very similar to routing a PCB, and I find it quite therapeutic as long as nothing goes wrong. Indeed, lots of testing was done throughout assembly to ensure everything worked and a little confidence soldering goes a long way.

Interestingly, this robot actually took me longer to prototype and breadboard than



it did for me to build/free-form! Without code, any changes to the robot's behaviour must be made by physically altering the circuit, which is significantly easier to do on a breadboard than on the eventual free-formed robot.

SOLAR-CHARGING CIRCUITRY

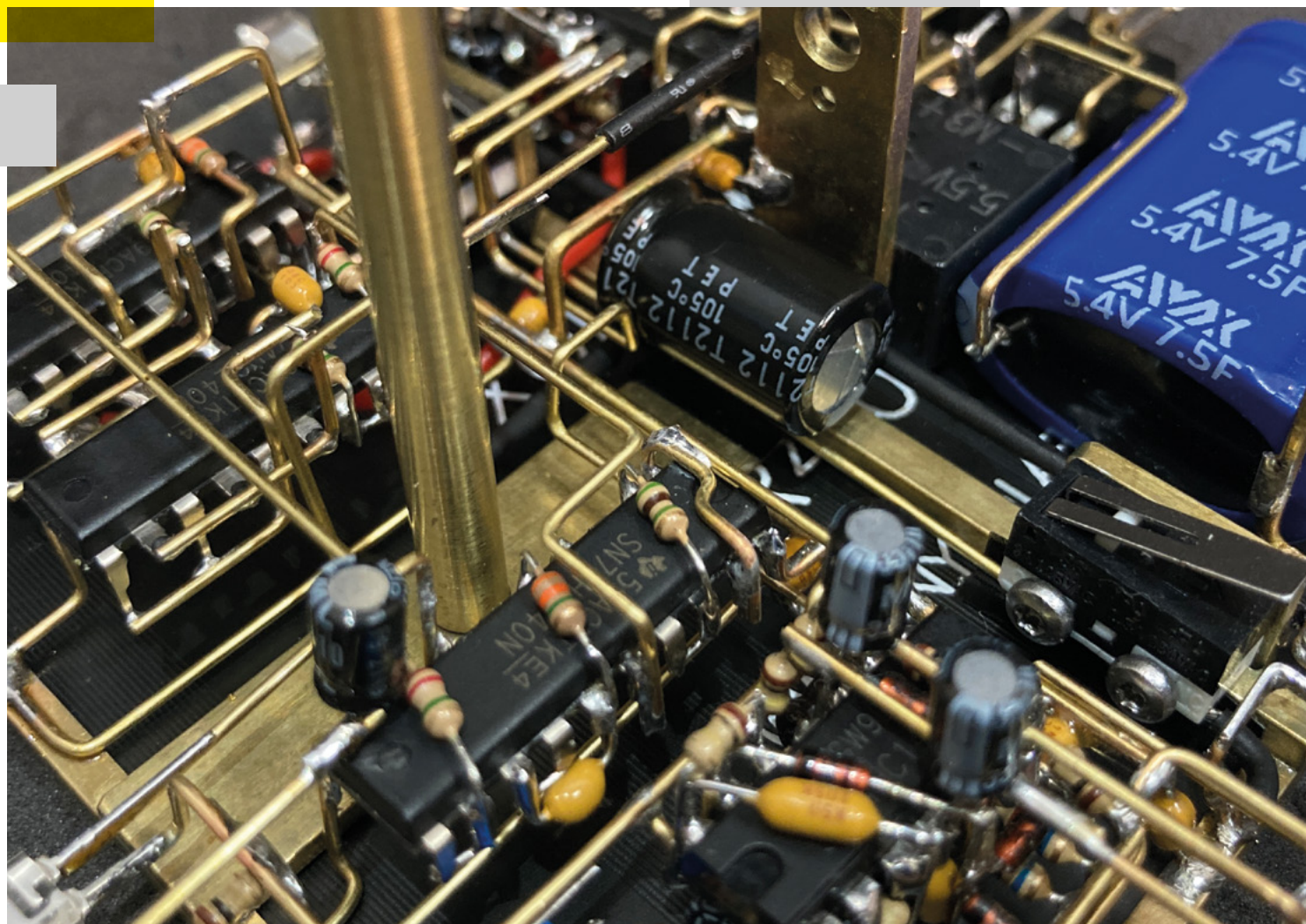
Starting the circuit explanation with power, the broad goal is to be able to run the robot solely off the solar panel. However, if you have ever played around with solar panels before, you might have noticed that they are immensely less efficient when out of direct sunlight and I wanted this robot to live on my desk all year round. The solar panels generally found on indoor devices like calculators are amorphous panels. These are different to the polycrystalline and monocrystalline panels used outdoors, and output lower power for their area. The ANYSOLAR panel used in this project is a



Above
The frame before free-forming

high-efficiency monocrystalline panel that boasts up to 25% efficiency and, more importantly, responds to a wide enough range of light to be effective indoors. Even still, the datasheet says to expect anywhere from 100–500 times less power when used indoors. Evidently, power is a big hurdle, but luckily we have plenty of BEAM tricks we can use in order to squeeze everything we can from this panel.

Indoors, the panel outputs $<100\mu\text{A}$ on a cloudy day and 5–10 mA on a sunny day, but the motors draw around 20 mA each! Clearly, the panel alone would never be able to power the robot, which is a problem that many solar-powered BEAM projects face. This problem is solved with a ubiquitous BEAM circuit known as a 'solar engine'. The goal of a solar engine is to allow the panel to charge a storage capacitor until there is enough energy to do useful work, at which point the solar engine turns on the rest of the circuit. Simple solar engines are robust and easy to implement, but are not always the most efficient. Hence, I have used a clever Wilf circuit, dubbed the Dual Slope Sampling Solar Engine (DSSSE). This DSSSE samples the storage capacitor voltage for only a short moment every five seconds or so and only draws significant power during that →



sample time. The solar engine on this robot turns on at around 3.5V and turns off again at around 2.5V. When off, only the charging is connected, and when on, the motor controller and microphone circuit turn on alongside a blue standby LED.

The DSSSE actually charges two capacitors: one is a smaller 330 mF supercapacitor and the other is a larger 7.5F supercapacitor. The circuit starts by charging the 330 mF capacitor so that, in the morning, it takes much less time for the robot to initially turn on and face the sun. Once locked onto the sun, the panel begins charging the 7.5F capacitor until it matches the 330 mF capacitor voltage, and the two capacitors act as one.

MOTOR CONTROL CIRCUITRY

The motor control circuitry is really elegant and was also (surprise, surprise) originally designed by Wilf. He called this circuit the Power Smart Head (PSH), and one PSH is used for each of the robot's axes. The PSH uses a modified CMOS oscillator to create what I would describe as a PWM-based, proportional controller, with a large dead-band. The oscillator has two resistor/capacitor (RC) pairs. The first RC pair dictates the oscillator frequency, and the duty cycle of this oscillator is altered by a pair of photodiodes that act as the robot's 'eyes'. This creates the PWM. The second RC pair creates the dead-band so that the robot does not just constantly move and

waste power. The interesting thing here is that from controller input (the voltage between the eyes) to the motor output (the PWM duty cycle), the signals remain entirely analogue!

WHISTLE/DANCE CIRCUITRY

The microphone circuit is based on yet another Wilf design that he called 'BEAM Sonics'. It consists of preamplification, followed by some BEAM-style logic, again mostly made of inverters and RC pairs. This logic ensures that the whistles are of the correct length, and that there is a distinct pause between each whistle. Once it

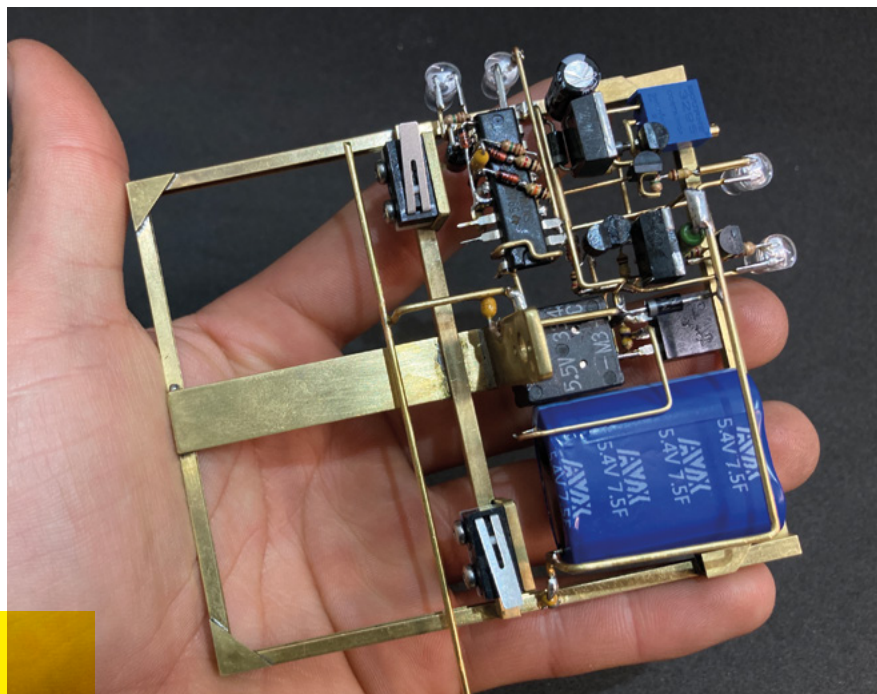
"THE SECOND RC PAIR CREATES THE DEAD-BAND"

counts three suitable whistles, it triggers a circuit to make the robot dance and another to lock the whistle circuit for 20 minutes so that the robot is not dancing too often and wasting power. The microphone circuit has three LEDs: one green that flashes for every whistle, one amber that flashes when the circuit activates, and one red that remains on while locked.

The dance circuit is simply three oscillators, two of which force the motors to oscillate and the other to make a pink, hexagonal LED flash.

CLOSING THOUGHTS

While BEAM circuits may take a little more effort to build and are nowhere near as powerful as a microcontroller, I would encourage anyone interested to do some research, and give these old BEAM circuits a look (the Wayback Machine might be your friend here). If you are keen to build any, I think that the best circuits to start with are 'pummers' that charge during the day and slowly flash at night, and 'photovores/photopoppers' that hop about using two

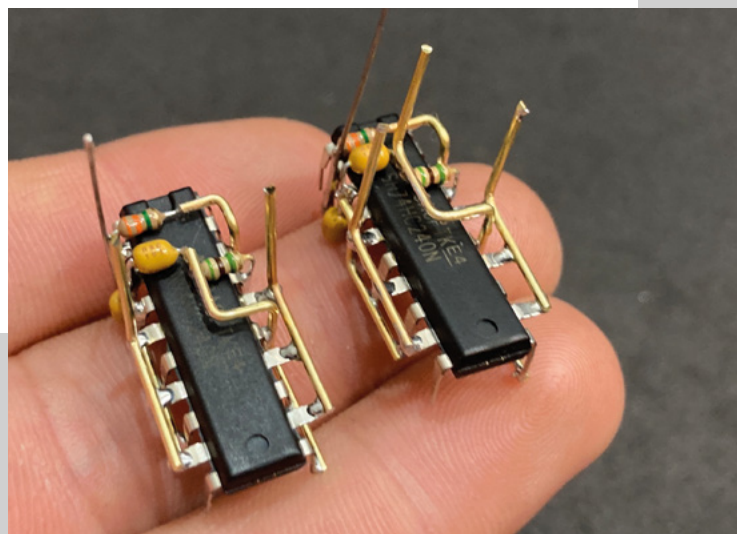


motors to locomote toward light. Although the obvious solution is often to slap a microcontroller onto projects, there may be times where simpler analogue circuits make more sense. In

fact, in recent years, analogue computers are once again being considered for use in artificial intelligence!

Even without building a BEAM solar engine, the principle of storing charge from a solar panel in a capacitor until there is enough energy to do something useful can be applied to many solar-powered projects. I hope this article might have inspired you to consider solar power for future projects, or perhaps to try free-forming, which can be a fun way to bring a little more artistry to any project. □

Above ↑
The completed solar logic



Right ♦
The two Power Smart Heads

HackSpace magazine meets...

Andrew Sink

3D printing professional, evangelist, and GoldenEye fan

Andrew Sink is a shining evangelist for 3D printing. He was there during the 3D printing boom of 2014, and kept the faith while the buzzwords died down and people refocused on what the potential of the technology was. He's written software to make design for 3D printing more accessible to new users, and his work is on the shelves of big retailers, without you even realising it's 3D-printed. He's on a mission: to bring 3D technology to the world in a way that people can understand intuitively. We're very lucky to have got a few minutes' worth of his insights to share with you here. →



Above  Andrew Slnk, wearing a mask of his own face. Thankfully this is the version without makeup

HackSpace: You're obviously not just a guy who uses 3D printing from time to time: you're an evangelist. What excites you about 3D printing?

Andrew Sink: So, first of all, in my day job, I am a senior applications engineer for Carbon (carbon3d.com). Carbon is a 3D printing technology company based outside San Francisco. So my job is primarily interfacing with customers and designing parts for additive manufacturing specific to the Carbon process. I work on a team that's primarily involved with lattice structures, so I spend a lot of time thinking how to design something that used to be a uniform block of material, as a conformal lattice, rather than a solid shape.

I was just at an expo in Detroit, Michigan, and I actually brought a bicycle with us to our booth. And it's got a 3D-printed seat, which was designed by people on the team that I work on. And it's designed for mechanical response; it's designed for comfort; it's designed for printability.

Thousands of pages of thought have been dialled into this lightweight bike seat, and then you look at it and go 'Oh, that's pretty neat'.

But what's really exciting about working at Carbon for me is I went into a bike shop to pick up that seat fairly recently. And it occurred to me that is the first 3D-printed part I've seen on a store shelf, that was a retail product. And when I was in Detroit with my wife, for the expo, we stopped in at a Dick's Sporting Goods [a big sports shop in the USA]. And they had the Adidas 3D-printed midsoles, so they had shoes that have a 3D-printed sole. And so I bought my wife a pair. It's cool to be able to walk into a store and buy something that was 3D-printed.

HS: What got you into 3D printing in the first place?

AS: I've been involved in additive manufacturing for about a decade now. I first used a 3D printer when I was in

college; we had one in the engineering lab, and I saw it and instantly felt that this is the thing that I'm going to do for the rest of my life – it was such a very clear and amazing technology.

The applications were just immediately apparent. I was taking a class on SOLIDWORKS as a 3D CAD program at the same time; taking a model out of SOLIDWORKS, and then sending it to the printer and then holding it in my hands that same day was an absolute revelation. It absolutely just tied together all the work that I was doing in the digital world to the physical world. And it was immediately apparent to me that it was just such an untapped technology.

I saw it and instantly felt that this is the thing that I'm going to do for the rest of my life

HS: And do you still have that same sense of wonder now that you use it day in, day out?

AS: I can't walk by a 3D printer. I'll stop and watch them for a couple of minutes. My house is filled with printers. It's just fascinating to see the industry evolve so rapidly in ten years – materials, software, hardware. And it's just, it's a really cool experience.

HS: You've also been experimenting with photogrammetry, which goes one step further – taking an object from the real world, putting it into the computer, then taking it back out again. What are the applications for that?

AS: The application of photogrammetry is hard. You could be sitting in a room looking at the first camera and saying, 'Well, what are we gonna do with it?' It's

such a limitless technology. And it's also very much in its infancy, and we're still figuring out what it's good for.

One of my favourite books is *Timeline* by Michael Crichton. It doesn't discuss 3D printing directly, but it talks about people who accumulate these things called transcription errors, where they're essentially turned into digital data streamed into a different area, and then re-materialised as physical objects. And over time, veins don't line up all the way, bones are slightly misaligned... you accumulate these errors over time. When I started getting involved in 3D scanning, I just immediately thought of that book, because it really breaks down a lot of these

core concepts really well. With photogrammetry, the thing you're scanning is going to be the thing you've scanned, but it's going to be different. It can be its own, you know, tangible item. The questions you get asked all the time with 3D scanning are, 'how do I scan a boat?' And that's a very different workflow from 'how do I scan a screw', where you're talking about precision on the magnitude of

microns versus metres. 3D scanning, just like 3D printing, means a lot of different things to a lot of different people. So, where I spend a lot of my time is on the accessibility and early entrance, so I think a lot about somebody who's never heard of 3D scanning before. They're going to go to Google and type in '3D scanning'; what's the first thing that they're going to see? And how is that going to be applicable to them? So I've spent a lot of time making videos showing the process of photogrammetry. How do you stitch photos into a model? How do you use a LiDAR scanner to use sensors to detect how a model exists in space, and really try and break it down in a way where somebody with no practical experience can solve their particular problem?

HS: Photogrammetry sounds like something that's really expensive to get into. Do you need to have access to a lot of specialist gear? →



Above 
Incredibly, this isn't
a drawn model,
it's a scan of the
dragon shown on
page 58

AS: Historically, it's been very expensive. But that's because it's been computationally expensive. The hard part is the software. That's where the industry has really lagged behind. A big part of that is conflicting messages from users, you know: I want to scan really small parts, very detailed, and I also want to fly a drone around my house and make a 3D model. Those two use cases are going to accumulate different amounts of data, and those are going to be processed differently.

Right now, there are apps on the App Store that are under \$10, that you can use on pretty much any iPhone, a lot of Android phones, and create a 3D model within minutes. There's this really great picture – my wife and I were in Chinatown in Boston. And it was a perfect day for 3D scanning – it was overcast, so the light was very diffused – typically, you get really harsh directional light from the sun, which makes it hard to scan stuff without adjusting the aperture settings on your camera. And it's just, it's kind of a mess.

We got this picture taken with us. I saw the statue right next to us, and I was like, 'Oh, that would be a great 3D scan'. And it is, without a doubt, one of the best scans I've ever made; you can make out individual teeth on the sculpture. This technology is so accessible now that it took me maybe ten minutes to walk around and get the photo set, and then maybe another five minutes after it finished rendering to clean up the base. So, photogrammetry has absolutely gotten to a point where it's affordable, it's usable, and you can get up and running very quickly.

For photogrammetry, it's really hard to get pictures of shiny stuff: bronze, for example, is terrible. Because it goes between dark and very bright, there are almost no gradients in bronze. You're either looking at, like, gold, or you're looking at black. And so a marble statue on an overcast day is just perfect. It's worked out great. 3D scanning shiny objects is sort of like taking a picture of a

mirror: you don't really get a picture of the mirror, you just get a picture of everything in front of it.

HS: Did you use an iPhone when you made the mask of your own face?

AS: That was actually done using a fairly high-end industrial 3D scanner. A friend of mine did it while we were at a trade show and we had a bit of extra time. I sat on it for a while – what are you gonna do with a scan of your face? And early in the pandemic, a company launched a bounty programme on a facial recognition spoof. It was something like \$10,000 if you could beat their biometric verification, and I thought, well, I've got a really nice-looking scan of myself, it probably wouldn't be a stretch to try and beat this thing. That became my Covid quarantine project. I was at home. I had some time on my

This technology is so accessible now that it took me maybe ten minutes to walk around and get the photo set

hands. And so I thought, what would a normal person do? I will make a hyper-realistic mask of my face.

I was going back and forth to make-up stores to get all kinds of different products to try and bring out highlights in the model. I was putting paint on the lips and stuff. It was creepy. It was absolutely awful. It worked in the sense that Google Photos will recognise the mask as a picture of me. So it will automatically tag me, which is cool. I'm able to set up an iPhone with the fake mask, but I'm not able to unlock it, so, whatever Apple uses for face ID does recognise it as a face, but it won't recognise it as the same face twice. Google recognising it as a face though, that was a huge win.

HS: You have a YouTube channel where you talk about 3D technology; you've also written some software to make it easier to create 3D models, haven't you?

AS: Yeah, so I'm a big fan of low poly art. I grew up with GoldenEye, on the Nintendo 64. So, I have a very deep appreciation for doing more with less. You look at the textures of these models, and it's like a guy, and that guy has a, you know, a texture on him. So if you see a face, it's the eyes and the nose. And then he moves his head, you're like, oh, it's kind of a rectangle shape, you know?

One of the things I found was that there weren't a lot of really intuitive workflows for creating low poly art without using programs like Blender, which is very powerful, but not very intuitive, so it's very difficult for beginners. I set out with a goal to design a web-based app to make low poly models. So the URL is **lowpoly3d.xyz** – the whole thing is written in JavaScript. And you can basically upload a 3D model and then you can select the destination amount and remove a certain number of edges, and it collapses the model into a low poly model, which you can then download.

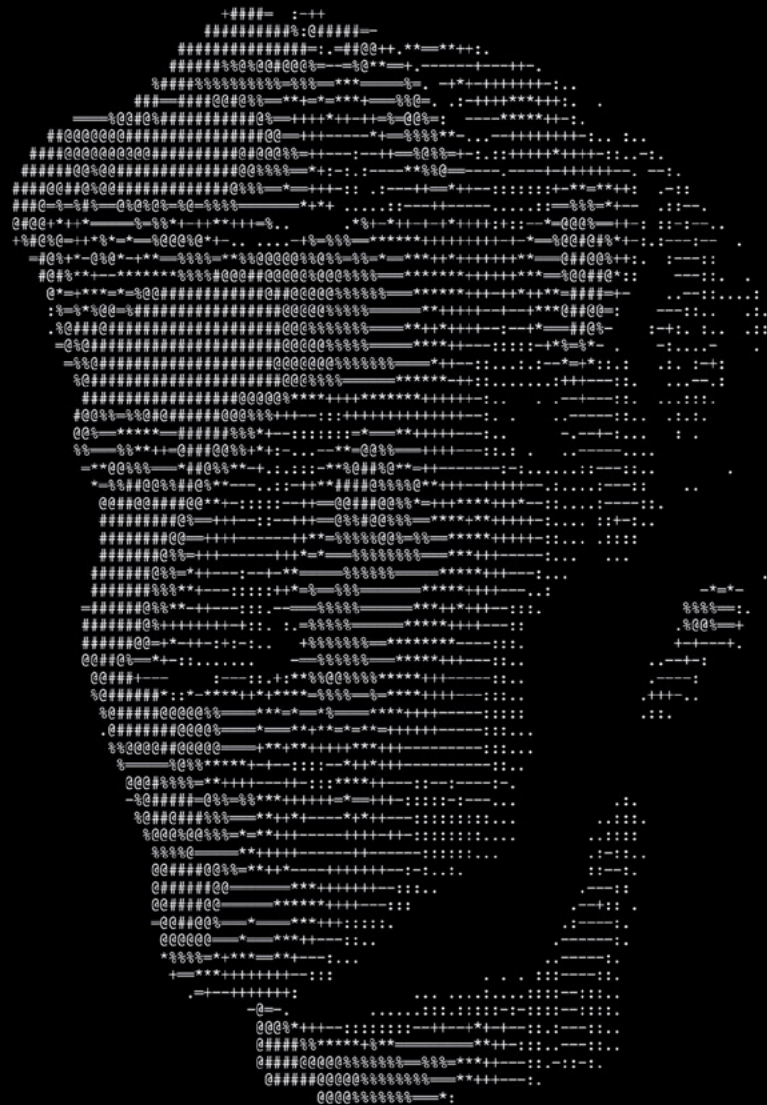
One of the reasons that I wanted to work on the low poly site was just because at this point, hardware has pretty well

outstripped software in 3D printing. We have very, very well-made machines that are made very inexpensively across a wide range of technologies. But, on the design side, you kind of have two options: professional parametric CAD, and then more sculpting programmes, like ZBrush, or Maya. Those both have pretty steep learning curves. So I wanted to create a tool that was easy to use, where somebody could say, 'Hey, I downloaded a model, and I'm going to reduce the poly counts. And now I'm going to print it out, and then it's done.' That was the goal. And I'm always excited to see what people are making with it.

And then there's also the STL to ASCII generator. Again, if you grew up →

[Download Text File](#)

Screenshot



Above 
Upload any 3D
image to Andrew's
andrewsink.github.
io/STL-to-ASCII-
Generator to
generate ASCII art

INTERVIEW

around the time that GoldenEye was out, you've probably seen ASCII art as well. You can upload a 3D model, and it'll basically apply a filter on top of that model that shows up as ASCII art. It's pretty cool. They're both designed to be as approachable as possible: you just drag a model and start hitting buttons, you really can't mess up. And I think that's very encouraging for beginners; maybe they've got a 3D printer for Christmas and they want to learn more about the modelling side, but they don't have access to expensive or difficult-to-learn tools.

HS: What do you see as the future, or possible futures, in the 3D printing multiverse?

AS: Let's split this into two separate parts. On the hobbyist side, I think hardware is

commoditised at this point. You can buy a printer that's mechanically sound for about \$200. That part's really easy, I think. The improvements will come in the form of ease of use and quality of life upgrades. So things like: can the printer tell you

"We're still not really at a place where the design software is matching that speed of improvement"

when it's out of filament? Can the printer detect when it's not actually printing? Those features are available right now on more expensive machines, and I think, as those become more prevalent, printers will become easier to use, and

consumers will be more incentivised to try them. There was this big boom in like 2014, where every newspaper had a think-piece like 'is this the dawn of 3D printing'? And it turned out that nobody knew how to design for 3D printing, so nobody was actually making their own stuff. For the last eight years or so, we've seen that hardware just continuously gets improved. A lot of the slicing software has seen incremental improvement, but we're still not really at a place where the design software is matching that speed of improvement. So I think on the consumer side, as those quality of life features start to make their way down, the less expensive printers and the software becomes more intuitive. I think that's going to really help drive adoption.

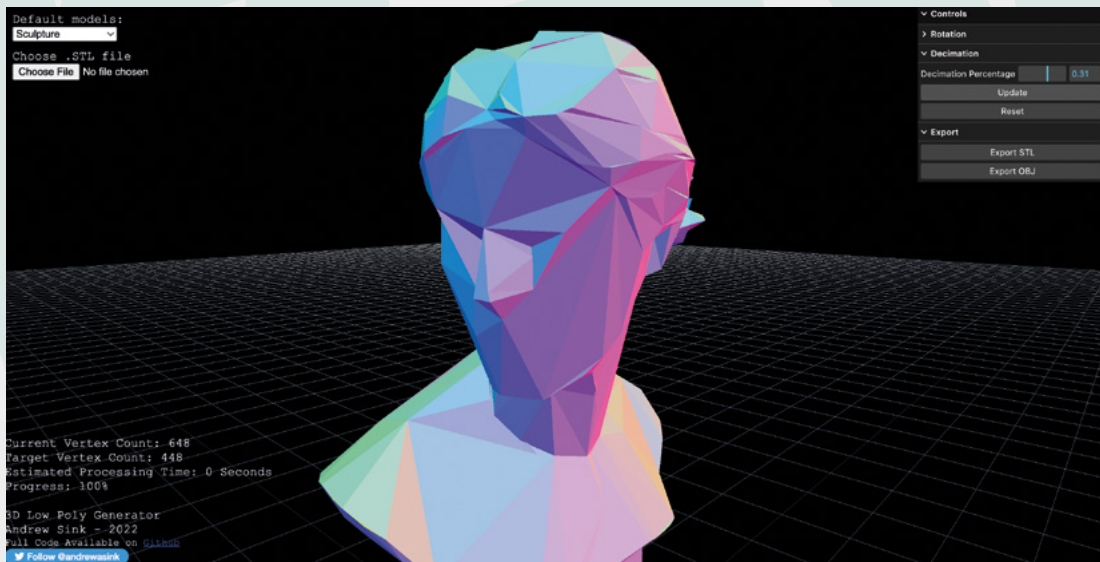
HS: And how about the industrial side?

AS: As the technology gets faster, the materials become more durable, and have better mechanical properties, you'll get to a point where you're going to buy a car, and your shifter knob is going to be 3D-printed, and you're not going to know it. Let's say there's a factory that makes cars, and they find out that one of their tail-lights – the bracket that they ordered 5000 of – doesn't fit onto the frame, and they need to print a shim. That is where 3D printing will come in and save the day, because you can solve this problem quickly, cheaply, reliably, without having to make tooling to mould this thing. I see a real future for 3D printing in solving problems that are below the surface, not necessarily in people buying things for the novelty of owning things that are 3D-printed.

HS: A lot of the time I see 3D printing projects online, someone's made the 1,000,000th Dungeons and Dragons or Warhammer miniature. I look at it and think I've seen this so many times before, and it's just the same as an injection-moulded thing, but less efficient. Do you think 3D printing is still exciting?



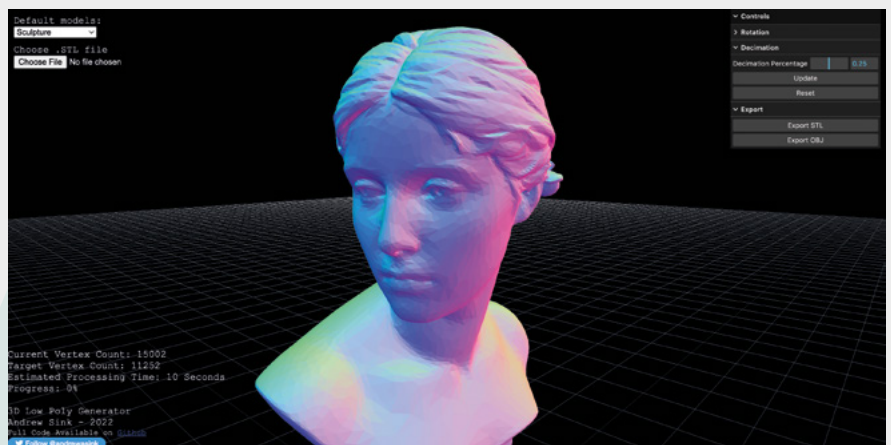
Right Compare the dragon shown here with the scanned image on p54; it's unbelievably accurate



Left  The same model in lowpoly3d.xyz, in high- and low-poly forms

AS: My Nana, my Italian grandmother, when she first saw a 3D printer, it was an old printer made of wood, and it was printing a fork. And she took one look at it and just immediately she was like, 'Oh, so you just draw something and it pops out'. Zero explanation required. It just immediately clicked in her head. And I thought that was just such a powerful thing. And so I try to make sure that, when I'm working on things like photogrammetry, or 3D scanning or the low poly generator, I'm thinking about what people are going to think when they see it.

Let me take you back to that first camera analogy I used earlier. Just like there are millions of little Baby Yodas and Dungeons and Dragons figures floating around, there's also millions of photos online of flowers, right? But people aren't going to stop taking pictures of flowers because it's already been done. There's something really uniquely personal about making something yourself. When you download something and print it out – it's an accomplishment – you made this. This is something you've brought into the world. I think that's a really special thing. It also leads to all sorts of things – maybe they get curious; maybe they learn CAD and go and create more things. Or, maybe they decided to become a photographer, and take pictures of flowers. 



Right  This disguise fooled Google, but not Andrew's iPhone





BARBED WIRE



Sharp, but pliable, barbed wire is ideal for practical indoor and outdoor reuse. Rosie Hattersley suggests getting creative



Rosie Hattersley

[@RosieHattersley](#)

Rosie Hattersley writes tech, craft, and life hacks and tweets [@RosieHattersley](#).

"IT BEGAN AS A GAME-CHANGING AGRICULTURAL MATERIAL THAT COULD BE USED TO FENCE OFF HOMESTEADERS' LAND"

If you want to seem 'alternative', introducing barbed wire into your decor is a handy short cut. The practical, low-cost means of demarcating your territory is visual shorthand for both unashamed attitude and keeping those you dislike at bay. No wonder it's beloved of warmongers and absolutists, as well as punks and edgy fashionistas. These days, specially produced barbed wire is often found artfully holding up displays of dried flowers rather than the original 'razor wire' type being deployed for more straightforward practical purposes.

Although we associate barbed wire with keeping people or animals at bay or interned, it began as a game-changing agricultural material that could be used to fence off homesteaders' land from their neighbours. In the US Midwest, where trees were either scarce or non-existent, being able to quickly

erect miles of fencing that wouldn't blow down in the prairie winds or need costly maintenance helped landowners demarcate their territory and prevent cowboys and cattle from ranging over and trashing their crops. Native Americans, chased off their own lands by fortune-chasers, were less enamoured.

Barbed wire was first introduced in the 1840s and 1850s, while Joseph Glidden, a farmer from Illinois, patented his version of barbed wire in 1873, improving on the version he'd seen at the DeKalb County Fair by twisting two strands of wire together for added strength and tension.

This more robust version of barbed wire also came in useful when the telecoms industry wanted to bring phone lines to rural communities, since they now had a cheap method of connecting to the telephone network using the wire fencing.

Electric and smart fencing has, to some extent, replaced the need for barbed wire in agricultural settings, but there are millions of miles of the near-indestructible stuff on farms and in former war zones. Depending on the regulations where you live, recycling such unforgiving material can be a challenge, and you'll find plenty of online advice about how best to store it and to unravel its deliberately tangled reels.



BARBED WIRE TRELLIS

Turning a former fencing material into a gate or decorative trellis might not seem a huge stretch, but there must be several hours' work involved in unravelling and then artistically bending and rearranging the spiky lengths of wire into a useful form. It's also arguably a more practical reuse than some of the crafty options we've seen. Dan Towell combines the attractive with the pragmatic with his 34-inch-tall trellises, designed to support and show off shrubs and climbing plants. The rusty, rustic look provides a good foil to effulgent foliage and is also the sort of idea a more whimsical gardener, keen to create a unique focal point, could consider emulating. →

Project Maker
DAN TOWELL
AKA THE
DUSTY RAVEN

Project Link
[hsmag.cc/
BarbedWireTrellis](http://hsmag.cc/BarbedWireTrellis)

Left ♦
There's probably some deep commentary in turning a utilitarian fence into an ornate trellis, but we just like the way it looks



RUSTIC PINE TABLE WITH INLAID BARBED WIRE

Project Maker
**JUNIPER
CANYON
DESIGN**

Project Link
[hsmag.cc/
InlaidBarbedWire](http://hsmag.cc/InlaidBarbedWire)

W

oodworking site Lumberjocks showcases interesting timber-based projects, as well as hosting a useful forum.

The site's mixed media builds include this beautifully finished table, made from a pine

slab that maker John, from Juniper Canyon Design, rescued and fashioned after discovering the miraculously rot-free stump of a tree that had fallen. He also just happened to have a roll of rusty barbed wire hanging up in his workshop. Just the right size for his uncle's horse trailer-cum-camper-van, the table was full of sap, making it tricky to work with. He used a belt sander with 100-grit sandpaper to bring out the wood's grain, before his wife set about routing a groove following the shape of the tree's outline for the barbed wire. "I pushed the barbed wire down into the rout groove and took advantage of the barbs by nailing them into the wood where possible, and adding a few finishing nails to any spots that wanted to pop up above the groove," John explains. He embedded horseshoes in the tabletop, in line with the table's intended use in a horse trailer.



Right ♦
Sometimes, all
you need is a
finishing touch



BARBED WIRE BALL

YouTube often comes up trumps for ideas about what to recycle and what you might reasonably produce from said scrap item. There are plenty of helpful video tips on how best to unfurl, stretch, and use (or roll up your unused) barbed wire rolls, but tutorials on how to achieve a barbed wire makeover



are scarce. Melinda's, aka Scrapmetalsheila, (scrapmetalsheila.com.au) excellent videos are an honourable exception. Armed with 100–200 m of barbed wire, engineering gloves rather than leather ones (since they catch on sharp objects), and a bolt-cutter, Melinda demonstrates how to create a beach-ball-size barbed wire ball. Having cut the wire into five 5-metre-long pieces, corralling the raw materials and tying it in place with flexible wire help you form the basic shape. After this, it's a matter of wrapping, securing, and tightening sections of wire.

Once complete, you could mount the ball on a rock or plinth in the garden, wrap it with solar lights, or dangle it from a light fitting and illuminate it. Should chandeliers and other lighting catch your imagination, there are lots of barbed wire versions to be spied in upcycling videos and blogs, but most omit how-to details. Scrapmetalsheila's advice will provide an invaluable guide.

Project Maker
SCRAP METAL SHEILA

Project Link
hsmag.cc/BarbedWireBall

Left ♦
Note – while this is beach-ball sized, we don't recommend that it's used as a beach-ball

BARBED WIRE EMUS

Absoriginal artist Laurie Nilsen's life-size barbed wire emu sculptures were inspired by him encountering a fence line that had ensnared more than 40 emus fleeing drought-ravaged plains, highlighting the material's dual role of both fencing off and fencing in. Laurie described having to put the dying emus out of their misery as one of the most gut-wrenching tasks he ever faced, but also acknowledged that he'd used barbed wire himself for fencing in former times. He often drew close-ups of barbed wire, calling attention to its dainty knots. Using satire to draw attention to social issues, several of Nilsen's emu sculptures are sited adjacent to places where

barbed wire is commonly used, such as prisons and law courts. Other artists who use barbed wire for sculpture and art also frequently have a personal connection with it, such as Ernie Poe, a former rancher who, at the age of 73, began the first of 200 sculptures each made from rusty barbed wire: hsmag.cc/WireSculptures. □



Project Maker
LAURIE NILSEN

Project Link
hsmag.cc/BarbedWireEmu

Left ♦
The emus are planning for a third victory, should there be another war with the Australian army

IN THE WORKSHOP: Beer BBQ

By Andrew Gregory

Combining two of our favourite things:
beer and barbecues



A

fter a butchery course a couple of months ago, I had a freezer full of meat; that meant I needed to upgrade my cooking facilities.

I wanted the option to barbecue, and I also wanted something with a top that I could close too, so that I could use it as an outdoor oven.

With this in mind, I trawled the internet for a nearby second hand beer keg that I could simply cut in half and stick legs on. Two candidates presented themselves: a stainless steel beer keg an hour and a half's drive away, or a barrel made of non-specific material, 50 minutes' drive away. I chose the nearer barrel. This was my first mistake. But we'll come to that later.

The barrels had been sitting outside a disused brewery, and were not rusty, but covered in old paint and something rust-adjacent. I set to work with a wire brush attachment on an electric drill to get the worst of it off, before marking out where I was going to cut. This is where the shape of the barrel gave me another problem: working out where the hinges would go for the lid. As there was only one flat bit in the middle of the barrel, I decided to cut it not in half, which would have been easier, but in two unequally-sized pieces, so that the sole hinge wouldn't have to take too much weight. I marked out the cutting lines, and set about it with an angle grinder.



Above ♦

We angled the leg attachments by a couple of degrees for more stability

Two days of struggle later, my neighbour, who is a welder by trade, asked me what on earth I was doing trying to cut aluminium with an angle grinder (here's where the initial error of going for the alloy barrel, rather than the stainless steel one, came back to haunt

I set to work with a wire brush attachment

me). Aluminium (and aluminium alloy, which is what it turned out my barrel was made of) has a very low melting point. Rather than cutting through the barrel, the angle grinder was cutting a very short way; then the heat from the cutting disc was melting the aluminium dust, which was cooling on the cutting edge of the disc, clogging it, damaging it, and eventually breaking it. A new approach was needed. →



Left ▣

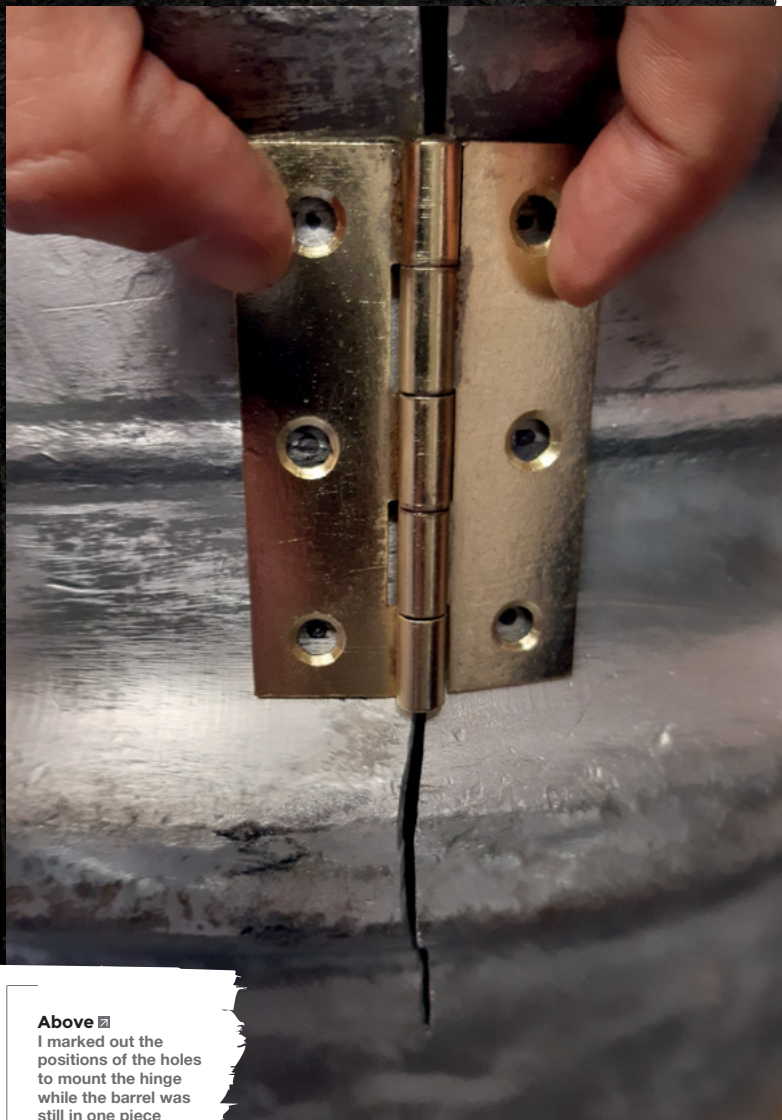
The reciprocating saw: looks like a Transformer (of the robots in disguise variety); actually, a terrifying cutting machine



Right ▣

Cutting off-centre made it easier to hinge

I had to go over the edges with a 40-grit sanding disc



Above I marked out the positions of the holes to mount the hinge while the barrel was still in one piece

I threw money at the problem, in the form of a reciprocating saw. This terrifying machine tore through the barrel in no time, apart from the rims, which are much thicker than the rest of the barrel, and which I cut by hand using a hacksaw. And, for the only time ever in my life, I managed to cut in a straighter line by hand than when using the power tool. A reciprocating saw cuts quickly, and so leaves a pretty rough edge on the cut metal, so I had to go over the edges with a 40-grit sanding disc – at last, a useful application for the angle grinder!

I'd seen a maker online who'd painted a beer keg to look like a can of Coca-Cola, leaving the ends unpainted to look like the polished aluminium of a Coke can. I liked the contrast between the painted surface and the raw metal, so I masked off the rims and applied many coats of Rust-Oleum barbecue and stove paint in British Racing Green.

Those rims, which look so good contrasted with the green, at this stage, became a blessing and a curse. A curse, because if I were to attach the legs to the relatively flat ends of the barrel, I'd have to cut gaps in the rims to allow the legs to sit flush. After the amount of work it took to cut through them the first time, I didn't fancy taking any more chunks out of them, so elected to attach the legs to the rims themselves, even though attaching a straight length of steel to a curved bit of aluminium isn't the neatest-looking solution.

Now, back to that initial choice of the alloy barrel over the stainless steel. Pure aluminium melts at 660°C. Aluminium alloy melts at... an unknowable



Above British Racing Green means we can grill faster



Above ♦

The highlights on the edge of this cutting disc are re-formed aluminium dust. Those bits don't cut anymore; they just get hot

temperature, since we have no idea what the aluminium is alloyed with. Charcoal burns at well over that temperature, at 1100°C, so there's a medium probability that, with the coals heaped at the bottom of the keg, the fire will melt through and drop onto whatever's below. I say medium, because the temperature of the air outside the barbecue will always be well below 660°C, even on a very hot day, so there's always going to be some air to cool the exterior – the worst that will probably happen is that it'll get a bit soft. To reduce the risk of even this happening, I bought a sheet of 2mm perforated steel and cut it to fit so that it sits roughly two centimetres from the bottom of the barbecue at its deepest point; steel melts at around 1300–1500°C, so as long as the hottest part of the fire sits on the steel mesh, all should be well. Cross your fingers for me. ▣

Danger!

Angle grinders work by spinning an abrasive disc at very high speeds. They throw up dust, sparks, and, if you're very unlucky, broken bits of grinding disc, which is why you should always use the right PPE when you're using one. For us, that comprised of the medical-grade face masks that we've had spare since Covid went away; a pair of padded work gloves; and a pair of safety goggles big enough to fit over the writer's extra-large spectacles. It's uncomfortable, it's hot, but that's nothing compared to the discomfort of getting a chunk of broken angle grinder disc in your eye.



Left ♦

From left to right: a cutting disc straight out of the packaging; a heavily damaged, unusable cutting disc; and a disc that got stuck in the barrel and fragmented



The
MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CustomPC

3 ISSUES FOR £10



FREE BOOK



hsmag.cc/hsbook

Subscribe to The MagPi, HackSpace magazine, or Custom PC. Your first three issues for £10, then our great value rolling subscription afterwards. Includes a free voucher for one of five fantastic books at store.rpiexpress.cc/collections/latest-bookazines
UK only. Free delivery on everything.

HackSpace
TECHNOLOGY IN YOUR HANDS

FORCE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG
74

TIME-LAPSE PHOTOGRAPHY

Videos for speed freaks

PG
78

for r in range(2):
 knit(20)
 addrow()

KNITTING WITH MATHS

Combining coding with yarn work

PG
84

K40 OPTICS

Make sure your mirrors are aligned

PG
90

INFRARED PHOTOGRAPHY

Photographs in the dark

PG
70

SCHOOL OF MAKING

Start your journey to craftsmanship with these essential skills

70 Welding helmet



PG
96

CAMERA PRINTER

Because photos should exist in the real world





Upgrade your helmet

How to be a cool welder on a budget



Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

A properly trained welder can earn a good living from their trade, and putting that money into good-quality equipment like a welding helmet is a sensible move.

For the weekend welder who spends more time holding a grinder than an electrode holder, it's harder to justify spending a large chunk of cash on an item like a welding helmet. In this article, you'll learn how a few simple helmet upgrades can make a big difference to your comfort without costing a fortune.

Welding can be an uncomfortable hobby. There are times when you'll find yourself hot, sweaty, and cramped into an uncomfortable corner, while you're struggling to see the joint you need to weld through your visor. Then, when you finally manage to get the weld, you need to raise your helmet and put on safety glasses to finish up with an angle grinder. It sounds miserable, but each of the problems mentioned here can be solved, and some welding helmets include solutions to these problems as a standard feature. Just because you're working with a basic model welding helmet, doesn't mean that

you should expect to be uncomfortable while you practise your welding skills.

Almost every welding helmet these days is auto-darkening. The lens appears to be tinted like sunglasses, but it blacks out even more when you start to weld. It's a very convenient feature, but it doesn't always work as you want it to. The initial tinting of the lens can be so dark that it's difficult to see the parts you're trying to weld clearly, and you either need to remove the helmet to double-check everything, or try to balance a flashlight somewhere to give yourself some extra illumination. Adding a work light to your welding helmet eliminates this problem completely, so it's an ideal place to start modifying your helmet.

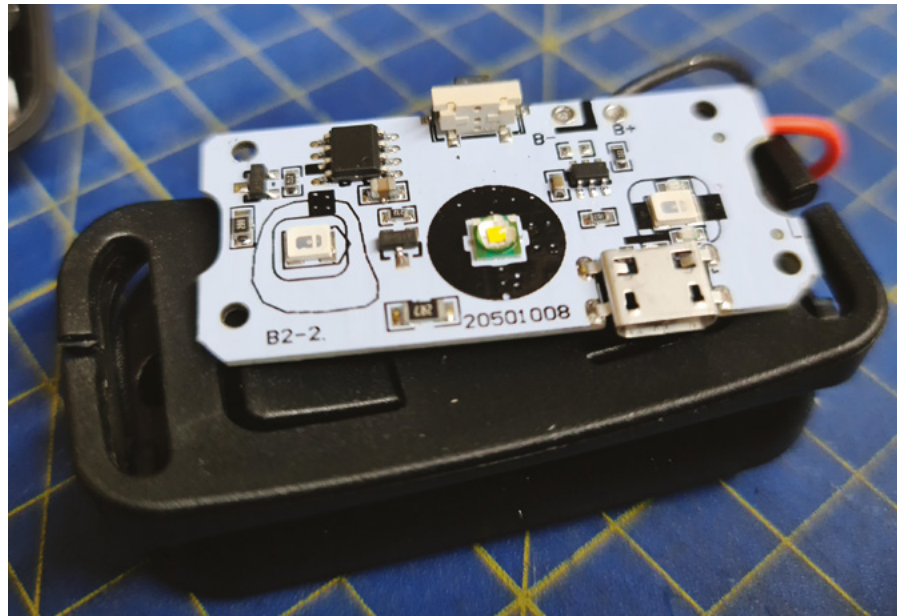
HACK YOUR HELMET

You can buy LED headlamps from almost every supermarket, but they're not a perfect solution for welding. The lamp needs to fit onto the helmet and not to your head, so the strap provided is pretty much useless. A bigger problem is that the lamps aren't designed to be used by someone wearing heavy welding gloves, and the tiny power button on most lamps is pretty much useless. On the plus side, the lamp has a built-in power source and, if you



Above

The elastic has been removed from the headlamp bracket, and the bracket has been bolted to the body of the helmet. A rivet, glue, or hook-and-eye fasteners could also be used to fix the headlamp in position



Above

You can clearly see that this lamp has a tiny PCB-mounted switch at the top that would be crushed into oblivion by chunky welding glove fingers. Next to it, on the right, are the main battery connections. These are the areas that you need to solder extension wires onto

can access it, then you can use it to power other helmet upgrades.

Crack open the LED headlight and take a look at the circuit board. You're looking for the battery power wires and the connections that lead to the button that turns the light on and off. Solder jumpers onto these wires and drill a small hole in the side of the light to pass these wires through. You should be able to remove the elastic strap and mount the light

//

Adding a work light to your welding helmet eliminates this problem completely, so it's an ideal place to start

//

on the helmet by screwing or gluing the adjustable bracket into a convenient place. Drill a small hole in the helmet and pass the wires inside, and then make a hole in the side of the helmet to mount a large arcade button. This arcade button will be your light's new on/off switch, so position it on whichever side feels most convenient. For a right-handed person, it's probably the most comfortable to position the switch on the left-hand side of the helmet because that's the hand that won't be holding a tool. Solder the switch jumper wires onto the arcade button, and use hot glue to hold the wires in place on the inside of the helmet. You should now be able to switch →

QUICK TIP

You might need to lower the brightness of the LED headlight to prevent it from triggering the auto-darkening feature of the helmet.

TUTORIAL



Above ♦
Blowers are available in a wide range of voltages and sizes, but something around a 50 mm 5V blower fan should fit into most helmets

QUICK TIP

Check the position of your light on the helmet and give it a quick test before you fix it in place. Check you're not blocking the recharge socket for the lamp.

Right ♦
If the case doesn't pull open with plastic spudgers, you might need to carefully cut the plastic seam using a sharp Stanley blade. Don't try to apply force with the blade, just make shallow cuts along the seam. Once you have a gap in the seam, switch back to using the plastic spudger and picks

the light on or off by tapping the switch on the side of your helmet.

The next modification for the helmet will help keep you cool, and will keep your visor (and glasses, if you wear them) free from condensation. Adding fans to the helmet is easy to do, but you don't just

want to use a standard PC or laptop fan. For one thing, the thought of spinning plastic blades near to your face isn't particularly comforting. More importantly, the welding helmet is designed to protect your entire face and not just your eyes – if you start drilling large holes through the plastic then UV light will be able to get in and give you a welding tan. A blower pulls in air from the front and uses an impeller to push air out of its side rather than straight through to the back like an ordinary fan. Blowers are generally more powerful than regular fans, and the enclosed blades and right-angled exhaust means that they're a great choice for a welding helmet. Your face is protected from blades and UV light. You can position the fans in your helmet so that the airflow



The thought of spinning plastic blades near to your face isn't particularly comforting



SPARE PARTS

You can buy replacement auto-darkening lenses quite cheaply, and even units with built-in grinder buttons and adjustable trigger settings aren't that expensive. However, if you've already got access to the parts you need and the modification only takes a few minutes to complete, why spend more money?

feels most comfortable to you. Faces and helmets are all different shapes, but try to get at least one fan blowing over the welding lens.

You can use the power cables that you soldered into the headlamp battery to power the fans, and all you need to do is add a suitable (chunky) toggle switch to turn the fan on or off. Positioning the toggle switch can be tricky on some helmets, but if

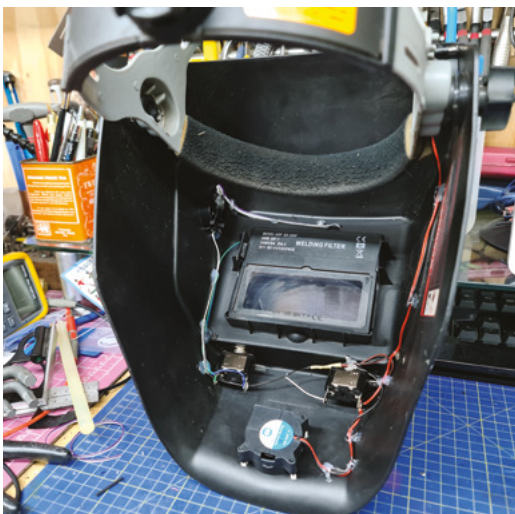


BATTERY PROTECTION

If your headlamp uses a LiPo battery, you need to figure out whether the battery is protected from excess discharge or not. If a LiPo battery is drained too much, it isn't possible to recharge it again and will need to be replaced. Some batteries have a discharge protection circuit built into the battery itself, while others rely on the attached electronics to deal with discharge protection. If you're not sure, err on the side of caution and add a battery discharge protection circuit to the fans.

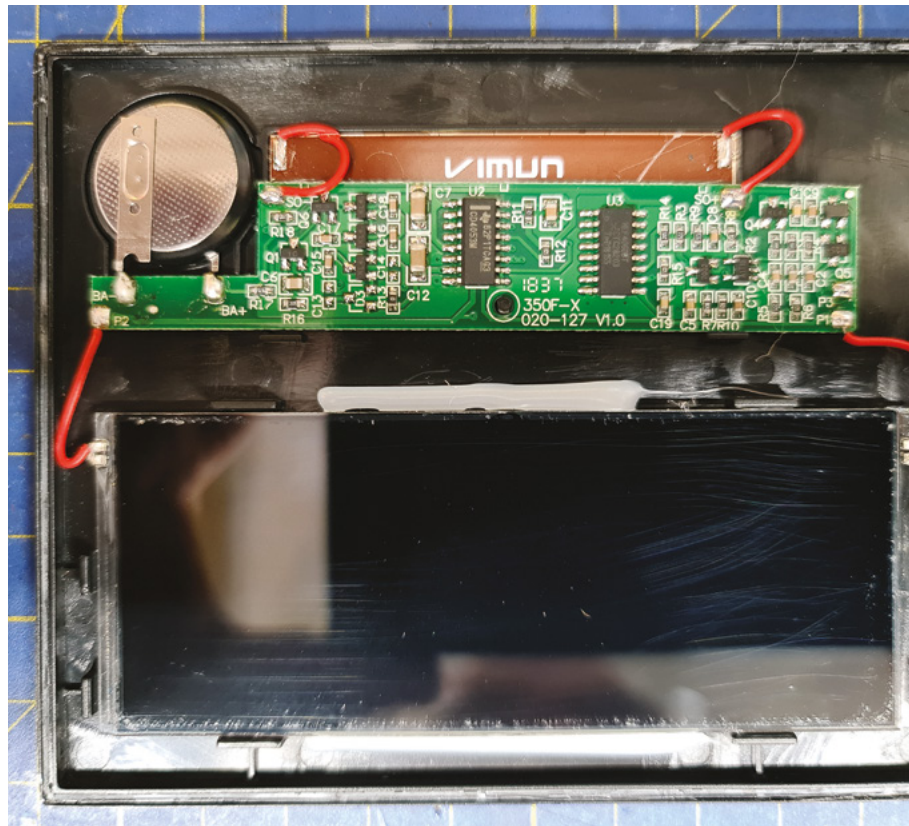
you can place it underneath the visor, you'll be able to reach the switch easily and toggle it while using the helmet itself for leverage.

The last modification for the welding helmet is the trickiest to get right, because it involves breaking open the auto-darkening lens and modifying it to add a switch. This means that you can stop the auto-darkening when it's not needed (such as when grinding), but ensure that you re-enable it before starting to weld. The circuit isn't complicated, but the housing for the auto-darkening lens is often glued shut and it needs special care to open it without damaging the plastic. Remove the lens unit from the welding helmet, and begin working along the edges of the case where the two halves meet with a plastic spudger and picks.



Above

The wires in this helmet are kept in place with hot glue. A thin layer of foam or epoxy resin will hide all of the electronics away permanently, although it isn't strictly necessary as long as there are no exposed wires or connections that could short out and cause a problem



DISABLE DIMMING

Once you have the case open and the circuit board exposed, you should see two wires running from the LCD screen to the circuit board. Disconnect the wire at the circuit board end, and solder on extension wires that lead to a toggle switch. Take a close look at where the helmet clips hold onto the lens, and make a hole in the lens unit's plastic case so that you can feed the switch wires out without interfering with the clips in the helmet. Toggling the switch to the open position will prevent the circuit board from altering the state of the LCD. Test the lens by shining a bright light onto the sensor. If everything is working correctly, reassemble the plastic case and hold the two halves together with a few dabs of superglue. Don't go mad and glue the whole seam, or you'll never be able to get the unit apart again if you need to. Just add a drop of glue at each corner, and an extra few drops along the edges to 'stitch' the two halves back together. The clips in the helmet will hold the two halves in compression, so the glue doesn't need to do any heavy lifting.

Keep all of the wires tidy inside your helmet with hot glue, and consider adding a layer of resin over the top of them to keep them firmly in place. Protect any exposed connections with hot glue or PCB lacquer, and you should be all ready to weld and grind without the uncomfortable side effects. □

Above

You can see the circuit used to control the auto-darkening lens, including the battery in the top left, and the connection that needs to have a switch installed at the point marked P1 on the bottom left of the PCB. It's easier to just cut this wire and extend it than it is to disconnect it from the board and replace it

QUICK TIP

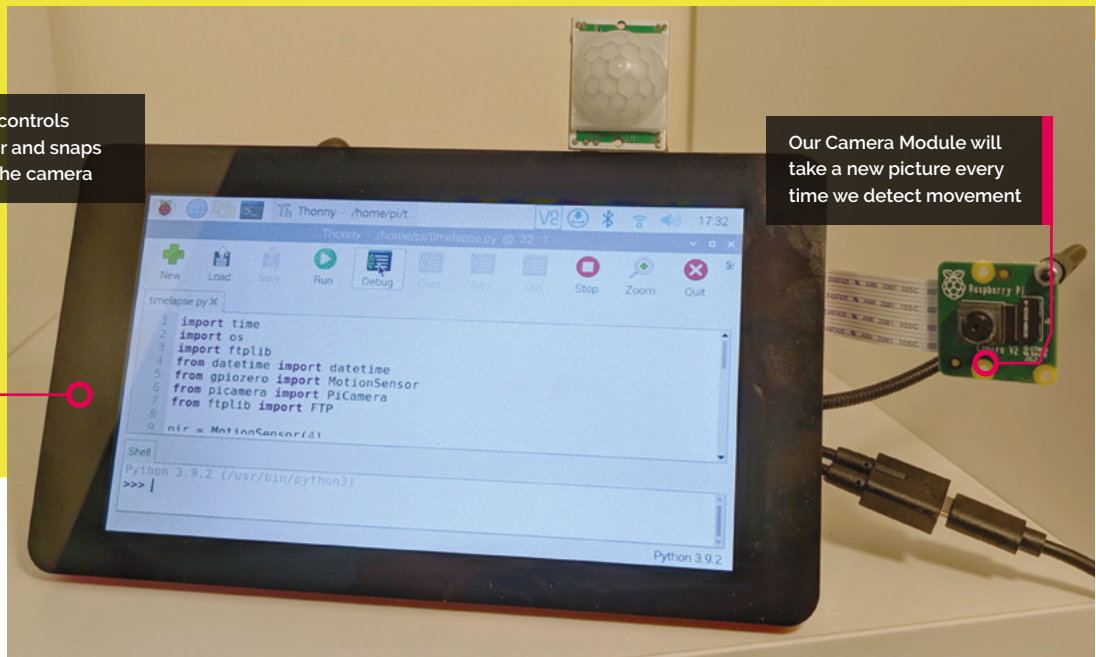
Beware that the welding helmet might not be safety rated for something like impact from a shattered cutting disc – always wear the proper PPE for the job you're doing.

TURN YOUR PHOTOS INTO TIME-LAPSE VIDEOS

Can we get to the good bit? Shooting time-lapse videos makes drawn-out processes more engaging

Python code controls the PIR sensor and snaps images with the camera

Our Camera Module will take a new picture every time we detect movement



MAKER Nik Rawlinson

Coffee-drinking, typewriter lover with a penchant for pencils and paint. Frequently found staring at the sea from the back of a camper-van.

nikrawlinson.com

Good things are usually worth the wait – unless they come at the end of a long and largely repetitive video. In this workshop, we'll show you how pairing Raspberry Pi with a Camera Module 2 and a PIR motion detector lets you shoot stills only when there's something worth watching. Discover how to convert images into a time-lapse video packed with action. We've been using this technique to show how time-consuming artistic processes result in a finished image, but you could also use it to see what your pet does when you've left it home alone.

01 Enable legacy camera support

The functions we need to use in our Python code will run natively on older versions of Raspberry Pi OS, but require legacy support if you're running Raspberry Pi OS Bullseye or later. To enable support in these versions, open a new Terminal window by pressing **CTRL+ALT+T** (or connect to your Raspberry Pi remotely using SSH) and type:

```
sudo raspi-config
```

Select option **3 – Interface Options** and **1 – Legacy camera**, then confirm that you want to enable legacy camera support. Quit `raspi-config` and allow your Raspberry Pi to reboot and implement the changes.

02 Correct your screen

We're setting up our time-lapse camera using a Raspberry Pi 3 B+ connected to a Raspberry Pi Touch Display (magpi.cc/touch). Enabling legacy camera support caused our display to rotate through 180 degrees (only locally; the orientation was still correct when connecting via VNC), and removed the Displays option from Raspberry Pi OS's graphical interface. If you experience the same, open a Terminal window and type:

```
sudo nano /boot/config.txt
```

Key down to the bottom of the file, add a new line, and type:

```
lcd_rotate=2
```

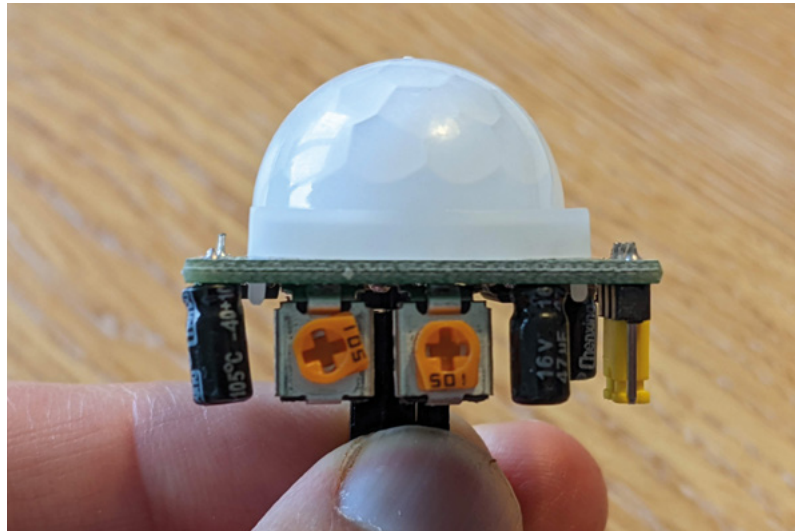
Press **CTRL+X** to quit and press **Y** when asked if you want to save, then reboot by typing:

```
sudo reboot
```

“ Enabling legacy camera support caused our display to rotate through 180 degrees ”

03 Connect your camera

We're using the regular Camera Module 2 here because we only want to shoot time-lapses in regular lighting. However, this tutorial works equally well with the Camera Module 2 NoIR, which can shoot low-light and night-time stills and video (you will need to provide an infra-red light source for night-time use). Connect the camera to Raspberry Pi's CSI (Camera Serial Interface) using the bundled ribbon cable. Don't confuse it with the display connector: on Raspberry Pi Model B computers, the camera connector is the one between the HDMI and headphone ports.



04 Connect your PIR sensor

The passive infra-red (PIR) sensor lets you take pictures only when there's some kind of movement. Place your Raspberry Pi on a flat surface so the USB ports are to the right and the GPIO pins are running along the top edge, and hold your PIR sensor so that its pins are also running along the top edge and pointing towards you. Use jumper wires to connect the PIR's left pin and right pins to GPIO pins 6 and 2 respectively, and the PIR's middle pin to GPIO pin 7 (magpi.cc/pinout).

▲ The screw on the right controls the sensitivity of the PIR motion detector

05 Position and test your camera

At this point, you want to get your Raspberry Pi and camera in position. We've mounted ours on a goose-neck stand, which plugs into an unused USB port, so we can point it directly at the scene we want to capture. Take a test photo by returning to the Terminal prompt and typing:

```
raspistill -o test.jpg
```

Open the resulting `test.jpg`, which will be saved in your home folder (`/home/pi/test.jpg`) to check the orientation, so you can apply correction later, if necessary.

06 Set up your FTP server

To avoid filling up your Raspberry Pi's microSD card, our code uploads each captured image to a server and then deletes the local copy. It doesn't matter whether you're running your own server on-site or using a remote commercial offering: what does matter is that you have FTP access. Log on to your host's control panel and create a folder for ➔

Top Tip

Zeroes ain't heroes

If any of your saved images are 0kB in size, delete them: FFmpeg may quit early if it encounters them.

You'll Need

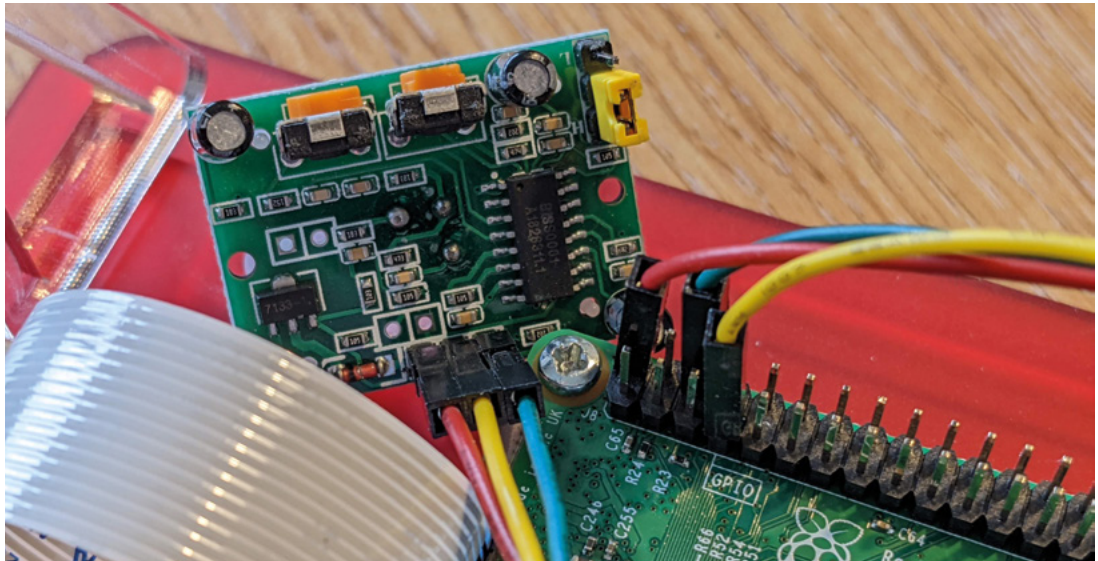
- Camera Module 2
magpi.cc/cameramodule2
- PIR motion sensor
magpi.cc/pir
- Three female-female jumper wires
magpi.cc/jumperjerry

Top Tip



Running too slow?

FFmpeg defaults to a framerate of 25 fps. To swap it for 60 fps, add '-framerate 60' before '-pattern_type' in Step 12.



uploaded images, as well as a user account that logs straight into that folder. The process for doing this will vary between providers. Make a note of your username, password, and server address.

07 Supply your login details

Download the code from GitHub (magpi.cc/timelapsepy) and save it in your home folder (`/home/pi/`) as `timelapse.py`. Open it in Thonny Python IDE (in the Programming menu of a standard install of Raspberry Pi OS) or an editor of your choice. Edit lines 21 and 22, inserting your server address where indicated on line 21, and your username and password where indicated on line 22.

▲ You'll need three jumper leads to connect the PIR motion detector to the GPIO pins on your Raspberry Pi

▼ If you're running Raspberry Pi OS Bullseye or later, use `raspi-config` to enable legacy camera support

08 Correct the camera... if necessary

If the test shot you took in Step 5 showed that the camera was rotated, uncomment line 17 by

removing the `#` at the start, and adjust the number at the end of the line to correct the framing. Our camera is positioned with Raspberry Pi's logo printed on the baseboard in the top left corner, so we need to rotate the image by 270 degrees to correct its orientation. However, you may instead need to make an adjustment of 90 or 180 degrees, depending on your setup.

09 Automate your capture

Save your edited code, then return to the Terminal and type:

```
crontab -e
```

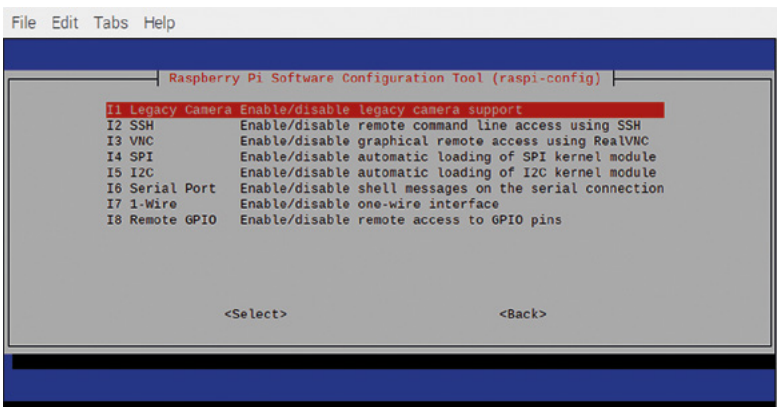
If you've never edited crontab before, you'll be asked which tool you want to use. Select 1 for Nano. When Nano opens, key down to the bottom of the file, create a new line and type:

```
@reboot python3 /home/pi/timelapse.py &
```

This tells Raspberry Pi OS to run your code as soon as it starts, with the `&` at the end telling it to run the process in the background. Press **CTRL+X** to quit Nano, confirm that you want to save your edit, and reboot your Raspberry Pi.

10 Testing, testing, 123

Your Raspberry Pi should begin capturing images and uploading them to your server when the sensor detects motion. If it doesn't, either reposition the sensor (not the camera) or adjust its sensitivity. To set the sensitivity to maximum, hold



the sensor so the dome is pointing up and the pins are away from you, then rotate the orange screw on the right (which controls sensitivity) as far as you can to the right, to maximise the range across which it detects motion. If you get false positives, dial back a little. Finding the optimal position can require some trial and error.

11 Download your stills

Download the stills from the server to a folder called **timelapse** inside your home folder. Open Terminal and type:

```
cd timelapse
```

Press **RETURN**, then type:

```
lv -v | cat -n | while read n f; do mv -n "$f" "$n.jpg"; done
```

Press **RETURN**. This renames your files sequentially. You now need to add padding so the file names are all the same length.

Type the following:

```
sudo apt install -y rename
```

Press **RETURN**, then type the following and press **RETURN**:

```
rename 's/\d+/sprintf("%05d", $&)/e' *.jpg
```

12 Create your time-lapse

Install FFmpeg by typing:

```
sudo apt install -y ffmpeg
```

Press **RETURN**. When installation completes, compile your images into a video by typing:

```
ffmpeg -pattern_type glob -i "*.jpg"
-s:v 640:480 -c:v libx264 -pix_fmt yuv420p
timelapse.mp4
```

If you want to change the output resolution, adjust **640:480** to your preferred dimensions. The time taken to complete the process depends on both the resolution and the number of images you've saved. □

File Edit Tabs Help

```
pi@picamera:~ $ crontab -e
no crontab for pi - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]:
```

timelapse.py

► Language: Python

DOWNLOAD
THE FULL CODE:



magpi.cc/timelapsepy

Credit: magpi.cc/repaddzeros and magpi.cc/renamingfilesseq

```
001. import time
002. import os
003. import ftplib
004. from datetime import datetime
005. from gpiozero import MotionSensor
006. from picamera import PiCamera
007. from ftplib import FTP
008.
009. pir = MotionSensor(4)
010.
011. def thegrab():
012.     thetime = datetime.now()
013.     detectiontime = thetime.strftime("%y-%m-%d-%H-%M-%S")
014.     extension = ".jpg"
015.     filename = detectiontime + extension
016.     camera = PiCamera()
017.     #camera.rotation = 270
018.     time.sleep(2)
019.     camera.capture(filename)
020.     camera.close()
021.     ftp = FTP('[server address]')
022.     ftp.login('[username]', '[password]')
023.     ftp.cwd('')
024.     ftp.storbinary("STOR " + filename, open(filename, 'rb'))
025.     ftp.quit()
026.     os.remove(filename)
027.
028. while True:
029.     pir.wait_for_motion()
030.     thegrab()
031.     time.sleep(30)
```




A stitch in code

The intriguing relationship between the fibre arts and coding



Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. She firmly believes that purchasing arts and crafts goodies, and actually sitting down and making something, are two completely different pastimes.

Above

Fibre arts have a lot in common with coding, as we shall explain

Have you ever considered how a knitting or a crochet pattern resembles coding, and the many parallels that exist between the fibre arts (an umbrella term which includes knitting, crochet, spinning, weaving, and yarn dyeing, to name but a few) and coding? These two seemingly unrelated fields have an interesting history together and, in this article, we will take a look at the fascinating similarities that really illustrate how much they have in common – spreading, we hope, an appreciation of the fact that coding and the fibre arts have many more similarities than you may ever have previously considered.

A CLOSE-KNIT RELATIONSHIP

Let's begin with a little history, which illustrates how coding/programming and the fibre arts are entwined. To see one of the most famous examples of this, we can simply look at the Jacquard loom, patented in 1804 by French weaver Joseph-Marie Jacquard. The Jacquard system was way ahead of its time, and enabled unskilled workers to weave intricate patterns easily: the central principle being the use of a

chain of interchangeable punched cards. In weaving, a single weft thread is passed over and under a set of warp threads. Before this invention, workers had to manually raise and lower warp threads in order to create any kind of fabric pattern, and this was an extremely slow process.

The cleverly designed cards of the Jacquard loom were fed into the Jacquard mechanism at the top of the loom and had small holes punched in them, which held the instructions for the weaving pattern. They basically controlled which warp threads should be raised to allow the weft to pass underneath. So, this was a very early example of instructing a machine to carry out an automated process, using a binary system (with either a punched hole, or no punched hole), where the information could be read by the loom and reproduced as required. The use of the replaceable punched cards to control a sequence of operations is widely held as an important step in the history of computing, as Charles Babbage (1791–1871) was inspired by the Jacquard loom's use of punch cards to program his Analytical Engine of 1837, considered by many as the first modern computer design. So, we can begin to see a link

GRAPHGHANS

There is a stitch in crochet called C2C or 'corner to corner', and each C2C stitch basically creates a small square or block. This means that C2C is a handy stitch to know if you want to design a pixel crochet blanket for example (also known as graphghans – afghans made from graphs), made up of a series of squares where each square represents one pixel of the image. You could write your own simple pattern for this on a piece of graph paper and then stitch it up. Or, you could use a website such as [Stitch Fiddle](http://StitchFiddle.com) (hsmag.cc/PixelCrochet) and create your chart there, where you can upload a picture to the site, or create a new design yourself. C2C is similar to how computers store images in grids of coloured squares.

What we are also illustrating here is that crocheters and knitters are very happy to use tech help, including websites and apps, in their designs if it makes the process easier – is your image of the average knitter or crocheter evolving at all? It's not just about the yarn these days...



Figure 1 ♦ The knitted swatch illustrates a simple lace design achieved by following the dictated code and various commands therein. In knitting, stitches stay on the needle and are active for a whole row, and you use either single point, double point, or fixed circular needles

- R2 – 3 Ch, skip first 3 Tr *[2 Tr, 1 Ch, 2 Tr] in 1 Ch sp, skip next 4 Tr, repeat from *, end by skipping last 2 Tr, 1 Tr in next Ch, turn
 - Repeat R2 until desired length
- (Abbreviations: Ch – Chain, Tr – Treble, Ch sp – Chain space, R – Row)

Figure 2 ♦ The crocheted swatch – a different effect is achieved from following the crochet code. In crochet, you complete each stitch one at a time, so only one stitch is live, and work with just a single hook

here between the two, on first sight, seemingly very different worlds.

ILLUSTRATIONS OF CRAFTING CODE

Let's now look more closely at the parallels between coding and the fibre arts, and perhaps the easiest thing to do to illustrate these is to take a look at some very simple patterns which, when knitted and crocheted, will produce the swatches shown in **Figures 1** and **2**, respectively.

SIMPLE KNIT PATTERN

- CO 16 stitches
- R1 – P2, [K5, P2] to end of row
- R2 – K2, [P5, K2] to end of row
- R3 – P2, *K2tog, YO, K1, YO, Sl1, K1, PSSO, P2; rep from * to end of row
- R4 – Repeat R2
- Repeat R1 – R4 until work measures 10 cm in length
- Bind off

(Abbreviations: CO – Cast On, R – Row, K – Knit, P – Purl, K2tog – Knit two stitches together, YO – Yarn Over, Sl – Slip, PSSO – Pass slipped stitch over)

SIMPLE CROCHET PATTERN (UK CROCHET TERMINOLOGY)

- Ch 15
- R1 – [2 Tr, 1 Ch, 2 Tr] in 5th Ch from hook, *skip 3 Ch, [2 Tr, 1 Ch, 2 Tr] in next Ch, repeat from * to last 2 Ch, skip 1 Ch, 1 Tr in last Ch, turn





Above ♦
We downloaded the free (open-source) pattern (code) from the designer (developer), channelled the Force through our hook and yarn, and hey presto... The Child! (hsmag.cc/BabyYoda)

rows of knitting or crochet. Note particularly, in both patterns, the use of brackets and asterisks, which are also of course used in coding – here, these are indicating a group of stitches that will be repeated, or a specific instruction to be carried out in a certain stitch such as, in the crochet example, completing two trebles, one chain, and two trebles in a single stitch. Designing a pattern is basically just like writing a program. You can think of the knitter or crocheter as the computer, translating the code, executing functions in the code line by line, in the correct order. Knitters and crocheters won't generally refer to these instructions as code; in fact, it may not occur to many of them that it is code, but any knitted or crocheted item, such as a scarf, is created one stitch at a time from such a piece of code. OK, if you're not a knitter or crocheter, these patterns won't make a huge amount of sense, and the purpose of this article is not to teach you how to knit or crochet. Also, knitting and crochet use very different techniques and tools to reach their end-products, but you can see from the patterns on the previous page that code plays a very important part in these two crafts – makers need the correct, error-free code in order to perform the function.

CRAFTY CORRELATIONS

So, looking specifically at the handcrafts of knitting and crochet, it can be suggested that the numerous comparisons between coding and these two fibre-based techniques include the following:

1. Standard notation

Knitting and crochet patterns are written using a standard notation describing precisely the actions required to achieve the desired outcome – a series of detailed steps to create an object or a piece of fabric made from stitches. Similarly, computer programs are also written using a standard notation describing the steps a computer needs to take to solve a problem.

2. Loop-the-loop

As already mentioned, if you take a look at the patterns on the previous page, you will see the use of asterisks and brackets. In knitting or crochet, these basically indicate the beginning of a stitch sequence which often needs to be repeated until you reach a

certain point, such as the end of the row, for example. Sounds familiar... a little like a loop in coding perhaps?

3. Terminology

Both the spheres of coding and fibre arts use their own languages or vernacular – a unique syntax that seems to make little sense unless you understand/ research the terms involved. A few baffling knitting expressions include rib, cables, brioche, intarsia, frogging, and entrelac, while crochet lingo includes seemingly unfathomable terms such as amigurumi, back post double crochet, filet, mosaic... and so on. But, as with anything, once you understand that jargon, it all falls into place.

4. Open source

In the world of the fibre arts, patterns are sometimes paid for, but can often be found for free. Fibre artists, like coders, are a generous bunch, and many people make their patterns available to all on sites such as Ravelry (see separate box). Hundreds of knitting and crocheting podcasters are also willing to share their knowledge for free on YouTube in videos showing

Note particularly, in both patterns, the use of brackets and asterisks, which are also of course used in coding

how to crochet a Baby Yoda, how to weave a wall-hanging, or how to knit a pair of socks. There is a real sense of community in the fibre arts world, as the many forums and online communities will attest – much like the world of programming and coding.

5. Maths, maths, and more maths

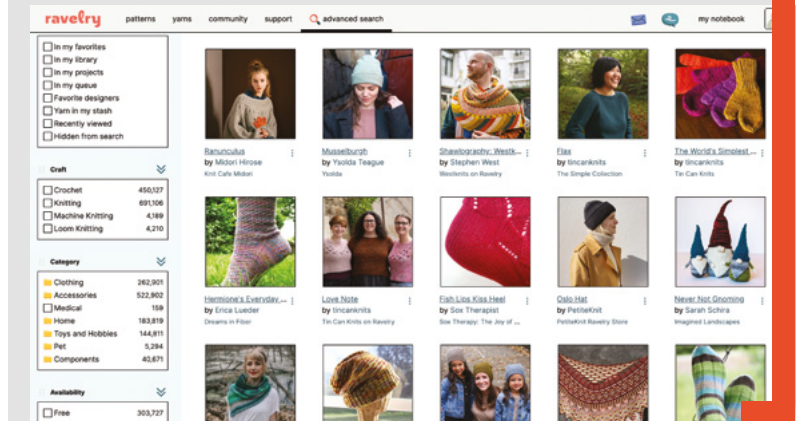
When this author is knitting and crocheting, at the same time, she is also counting. It's about order, accuracy, and precision. Following the code, she is processing that algorithm that a designer has created in order to instruct her on how to achieve the end goal. When executing the code, she needs to count (a lot!). Stitch counts are critical, so she needs to follow the designer's code precisely, or that crocheted version of Bernie Sanders at the inauguration will have the wrong-sized head, (yes, a talented individual had the ingenuity to design such a pattern – hsmag.cc/Bernie). 'Sssh, I'm counting' is a line that this author's family are more than familiar

RAVELRY: A CODING AND KNITTING COLLABORATION

There are many stereotypes associated with knitting and crochet but, these days, the average age of a fan of these two crafts, in particular, is much lower than in years gone by. Banish (please!) any thoughts of little old ladies in rocking chairs – fibre artists are cool, on-trend, tech-savvy, and want to regularly communicate with fellow creators online.

Consequently, tools have developed to meet the inherent demand for a form of 'social networking' specifically for the yarn arts. Ravelry (ravelry.com) was created in 2007 by Jessica and Cassidy Forbes from Boston, Massachusetts, and now has over nine million (yes, you read that figure correctly!) users worldwide. Interestingly, one half of the founding partnership was originally a knitter, and the other was... a coder! Their complementary skills are still used today to keep this gargantuan network functioning at its optimum. Nearly half a million knit and crochet patterns, for example, can be downloaded from the site, some free, some paid for, thus generating an income for thousands of pattern designers and sustaining Ravelry so that it can continue to unite fibre artists.

Users can utilise the organisational tools, share patterns, stashes, thoughts, ideas, discussions, and projects. It's a fibre-loving community using a massive database of patterns and yarns – you can filter your searches on hundreds of criteria and get instant access to exactly what you're looking for. It's free to join, and a great place to visit if you've lost your crochet or knitting mojo, as you're guaranteed to want to start at least 150 different projects once you start looking for inspiration.



with. Believe it or not, repetitive counting is also very calming as it's so meditative.

This author recently downloaded a free crochet pattern in which the maths didn't add up, based on what the code was telling her to do and the required stitch counts at the end of each row. The pattern appeared to not have been tech-edited, so she emailed the company concerned and rewrote the bothersome incorrect elements of the pattern. She's determined to write more of her own crochet code now, as it's really not that hard!

6. Refactoring and checking for 'bugs'

Just as coders will refactor code to try and improve the design of a piece of software, fibre artists do the same thing, as it's all in the finish. Most paid-for patterns, and many free patterns too, will be →

Above
A vast database of thousands of patterns at your fingertips, or should that be needle tips?

If you would like to experience the many therapeutic benefits that knitting, crochet, weaving, or any other fibrous form of creativity can offer (mindfulness, reduction of anxiety and stress, improved cognitive function, sense of control, sense of achievement... to name just a few), then there are many options:

- When you get that 'light bulb' moment, and everything finally clicks into place, you'll feel such a sense of achievement and you'll be away (oh, and start making space for some yarn, fibre, tools... it's a rabbit hole). You might even start designing your own patterns... also known as coding!

It's a given that coders require technical expertise, but fibre artists have those skills too. Being able to read a pattern and translate it is a technical skill, as is knowing how to knit or crochet in the first place, knowing what yarns to use for each pattern, measuring gauge correctly so that the new jumper you are constructing fits you properly, or modifying patterns to include your own personal slant, for example. (See 'Graphghans' box.)

**You can do both – you can
code and program and you
can knit or crochet or
spin or weave**

As in all areas of life, both coders and fibre artists are constantly facing challenges to develop, to learn more, and to hone their skills, as there are always new elements to master. Fibre artists will be learning new stitches, techniques, about new yarns, how to read charts, how to fix errors, while coders need to keep up with new languages, frameworks, libraries, and so on.

Perhaps a key point to make here is that you can do both – you can code and program and you can knit or crochet or spin or weave. You can be both creative and technical at the same time. Coding and the fibre arts are not mutually exclusive pastimes – would you be surprised to find a coder or developer who likes to knit, or a professional crochet designer who also coded for fun? Based on the many similarities discussed here, we wouldn't be surprised at all... and there are indeed many people who happily do both.

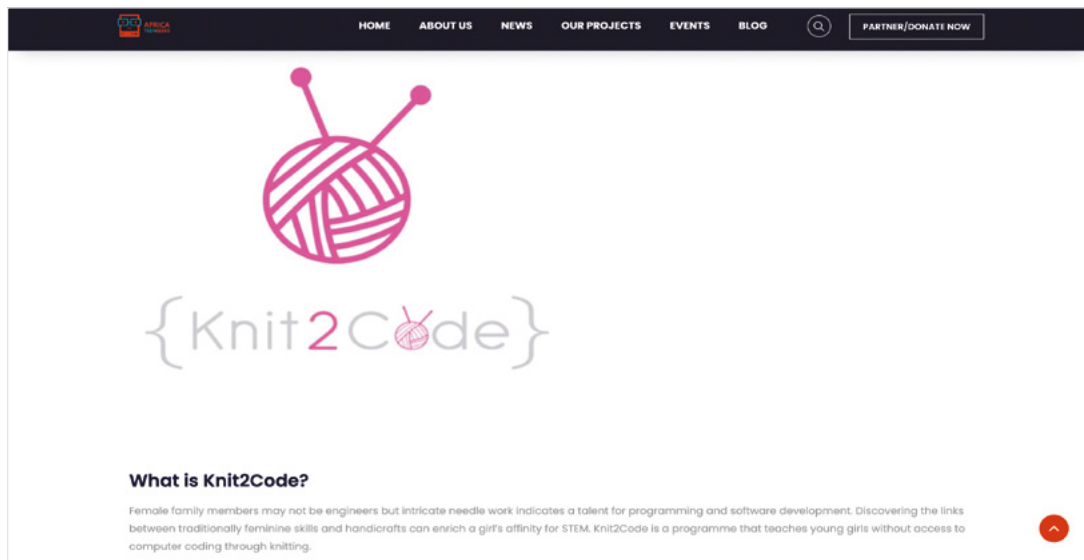
So, just to see what would happen, this author decided to take a very basic knitting pattern and turn it into Python code (with the help of her code-loving husband!)

- CO 20 stitches
- R1 & R2 – K
- R3 – K
- R4 – K3, P14, K3

rigorously tested by test knitters or crocheters, and by tech editors, to make sure that there are no errors in the pattern. Much like coders and developers, designers will search for bugs in the system – the end goal in a knit/crochet piece of coding is documentation that delivers what it promises, and only once it has been debugged can it be released, or deployed in software terminology. As in coding, pattern designers will use trial and error to achieve their desired outcome. Designers will rewrite their code so that they explain the pattern, in as few lines as possible, very similar to the DRY (Don't Repeat Yourself) principle of software development. Fibre arts patterns and coding are both trying to communicate complex instructions as simply as possible.

Fibre artists are, by definition, creative. They often tweak patterns, add their own touches, choose appropriate yarn colours, and basically express themselves through their yarny creations. Take, for example, yarn bombing (hsmag.cc/YarnBombing) – a creative, edgy, expressive form of street art, spreading colourful fibre joy. This author has recently found a pattern for crocheting toppers for UK postboxes, and hopes to create one soon for her local postbox. And, yes, coders are also creative – they need to possess a technical creativity in order to write any piece of software. It can be suggested that coders are aiming for a form of beauty in their code,





Left ♦ Africa Teen Geeks' Knit2Code programme brings together girls of different generations to learn, relearn, or enjoy knitting and, as a by-product, the basics of computing

- Repeat R3 & R4 until work measures 10cm in length
- Final 2 rows – K
- Bind off

You can find the link to the resulting Python code here (hsmag.cc/issue57). As you can see, there's quite a lot of repetition/looping here, along with a conditional ('until work measures 10cm in length'). In our Python code, we created functions for knit and purl, which simply add a letter 'k' or 'p' to a 'row' string the number of times specified in a 'stitches' parameter. We also created a special 'row4' function which calls the knit and purl functions to execute its mixture of knitted and purled stitches.

An 'addrow' function adds each completed row to a 'front' or 'rear' string – since in knitting, the side of the work is reversed with each row, we alternate this.

The first 'for' loop knits two rows (akin to 'R1 & R2 – K' in the pattern). A 'while' loop then repeats rows 3 (knit 20) and 4 until the 'length' variable (increased with each row) is 10 or greater. We finish with another 'for' loop to knit two more rows. At the end, we print the 'front' and 'rear' strings, which results in a chart of stitches for the front and rear of the work (**Figure 3**).

To conclude, if you'd like to read more, we really suggest that you take a look at this article from The New York Times (hsmag.cc/YarnProgrammable) where the ideas of physicist Dr Elisabetha Matsumoto, that 'yarn is a programmable material' and that 'knitting is coding', are explored. It's mind-blowing stuff, looking specifically at the mathematics of knitting. Here, we have really only just skimmed the surface of the coding and fibre arts entanglement, but hope we have piqued your interest in this fascinating area. □

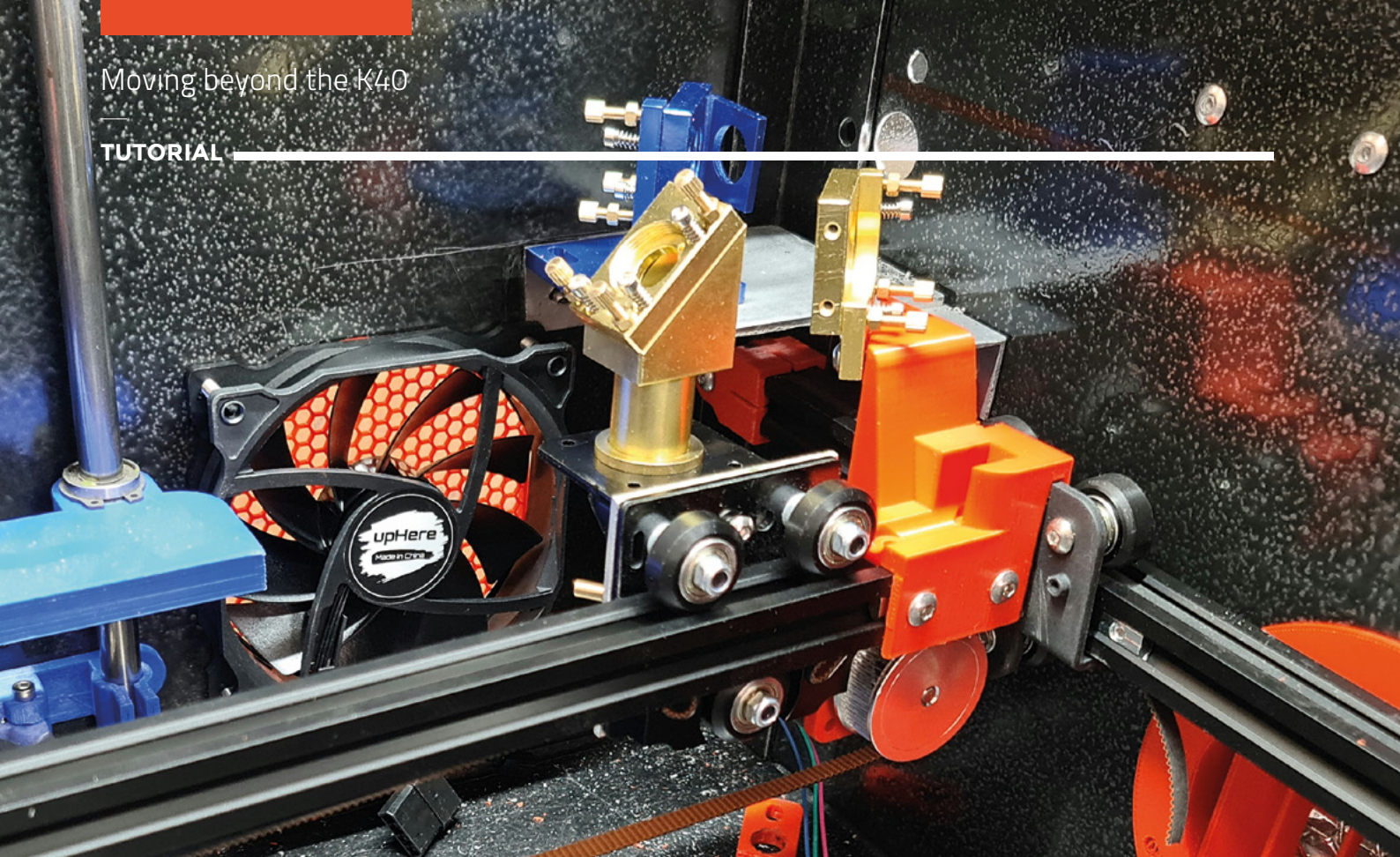
COVERT CODE

Did you know that you can encode secret messages into that hat or cardigan that you're knitting? Well, this is something that was used to great effect during wartime – yes, unassuming knitters became useful spies, using steganography (a way of hiding a message physically in plain sight) and passing on valuable information in their pattern of stitches to the people who could use the information effectively in the war effort. It was relatively straightforward to encipher information into a fabric using, for example, Morse code. When you think about it, it makes complete sense – knitting consists of two kinds of stitches, knit (flat stitches resembling a 'v') and purl (horizontal bumps) – and it is the chosen combination of these that makes a pattern, and is relatable to binary code.

To illustrate, during World War I, Belgian Intelligence agents sought the help of knitting women who could see train stations and tracks from their windows. They asked them to monitor the passing of German Imperial trains – in their knitting, these inconspicuous-looking women would drop stitches or add extra purl stitches and so on, as a form of predetermined code, to signify how many artillery trains had passed, if a train full of troops had gone through, etc. They then passed their completed fabric to the Belgian Resistance, who could decode the fabric, based on their pre-agreed stitch meanings. Had they been discovered by the enemy, these knitters' lives would, of course, have been in peril. It must have been a successful form of information gathering, as coded knitting played a part in WWII as well. This is a truly fascinating subject area if you are interested in history and knitting espionage, and you can read more here: hsmag.cc/KnittingSpies.



Above ♦ Who'd have thought that a humble ball of yarn and two wooden sticks could communicate vital intelligence?



Configuring your cutter

Dial in your laser cutter's settings



Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Above ♦ The bigger the laser cutter, the more difficult it will be to calibrate, because errors are amplified over the long distances

Even the most highly engineered products need to be configured correctly when they are installed, so it's no surprise that a homemade piece of kit based on a K40 laser will need some finesse and a few well-

placed hammer taps before it will perform properly. In this article, you'll learn some tips and tricks that will get your machine levelled, trammed, and generally calibrated in the safest way possible.

DON'T LOOK AT THE INVISIBLE LIGHT

The most obvious set of calibrations that you need to do with a laser cutter is aligning the mirrors. Ideally, this should be done after you've calibrated the gantry, or you'll risk having to start the whole mirror adjustment process again. Mirrors in the laser cutter are angled to bounce the beam into the lens regardless of the position of the laser head. Each mirror is around 20mm in diameter, so you need to be accurate. The difficulty with calibration is that the laser beam is invisible, and the mirrors change

position as the gantry moves. The typical advice is to use layers of masking tape or custom labels to see where the beam is striking the mirror, but this isn't very useful if you've only just put the machine together and the lasers aren't even close to aligned. A very slight change to a single mirror can shift the beam's path by several feet over a relatively short distance, missing any masking tape or focussing targets completely. Some expensive lasers use a beam combiner to inject a low-powered visible laser beam into the same path as the CO₂ laser beam so that it's easier to see exactly where the beam will hit an object, but these are typically an expensive extra that needs to be calibrated anyway.

A cheaper option for the home builder is to either remove the lens and fire a visible laser through the system in reverse, or remove the laser tube completely, and replace it temporarily with an appropriately mounted visible laser emitter. The latter option requires more effort to calibrate the laser, but the parts can be printed on a standard 3D printer and don't need to be particularly accurate. The former

option is neater and doesn't involve removing the tube, but it does require a lathe to create the parts with sufficient accuracy.

Print or make a suitable caddy to hold your visible laser emitter. Assuming that you are using a 3D printer, create the part so that the outer dimension of the caddy is the same as the measured diameter of the laser tube and the inner dimension holds the laser emitter firmly and squarely in position. Place the

“

The burn mark will tell you whether the beam is striking the mirror in the centre or not

”

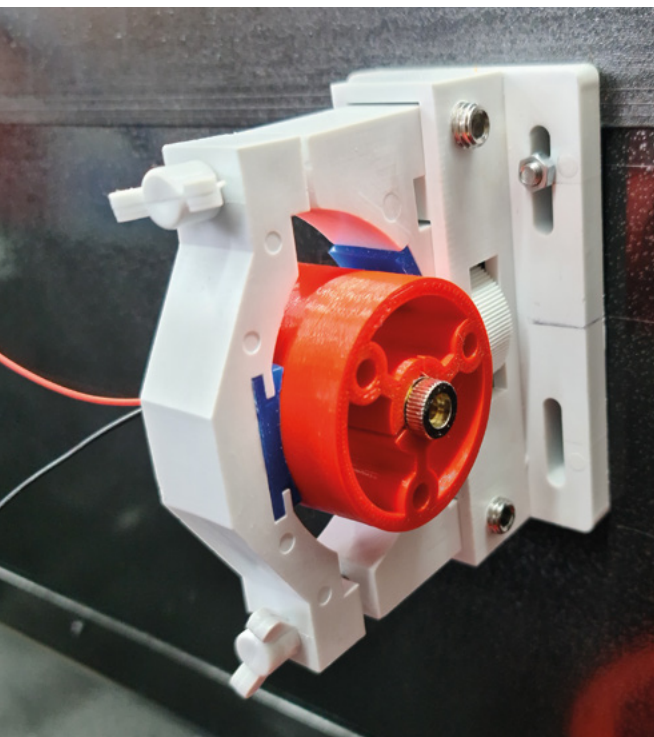
caddy into one of the clamps for the laser tube, and position it so that the beam is parallel to both the clamps and the base of the machine. You can do this by measuring the distance of the beam from multiple locations along the surface that the laser tube clamps are mounted to, and the base of the machine. Your emitter is aligned when the two measurements are consistent along the length of the machine. With this visible laser in place, it should be easier to see where

SAFETY

The initial setup and calibration process for a laser cutter is probably the most dangerous operation you can carry out on the machine. You will be operating a machine that might fire a laser at an unexpected time in an unexpected direction, or move motors in ways that you might not expect. The process of calibration takes time and requires you to perform the same series of tasks several times. It is easy to fall into a rhythm and forget the risks, and with repetition comes a strong temptation to disable some of the interlocks to make the process of calibration quicker. Do not disable interlocks unless it is absolutely necessary, because they are there to protect you from serious harm. Invisible laser radiation can disfigure or blind you. The power supplies inside a laser cutter can generate high enough voltages to cook meat. You are made of meat. Always wear laser safety goggles when working with the laser, and be aware of the risks from mechanical parts, high voltages, unexpected fires, and reflective surfaces. Never leave the laser in a state where it could be activated accidentally by someone who does not understand the risks. Always have appropriate safety equipment nearby.

QUICK TIP

Don't disable the interlocks and fire the laser with the chamber open, even if you're wearing appropriate safety goggles.

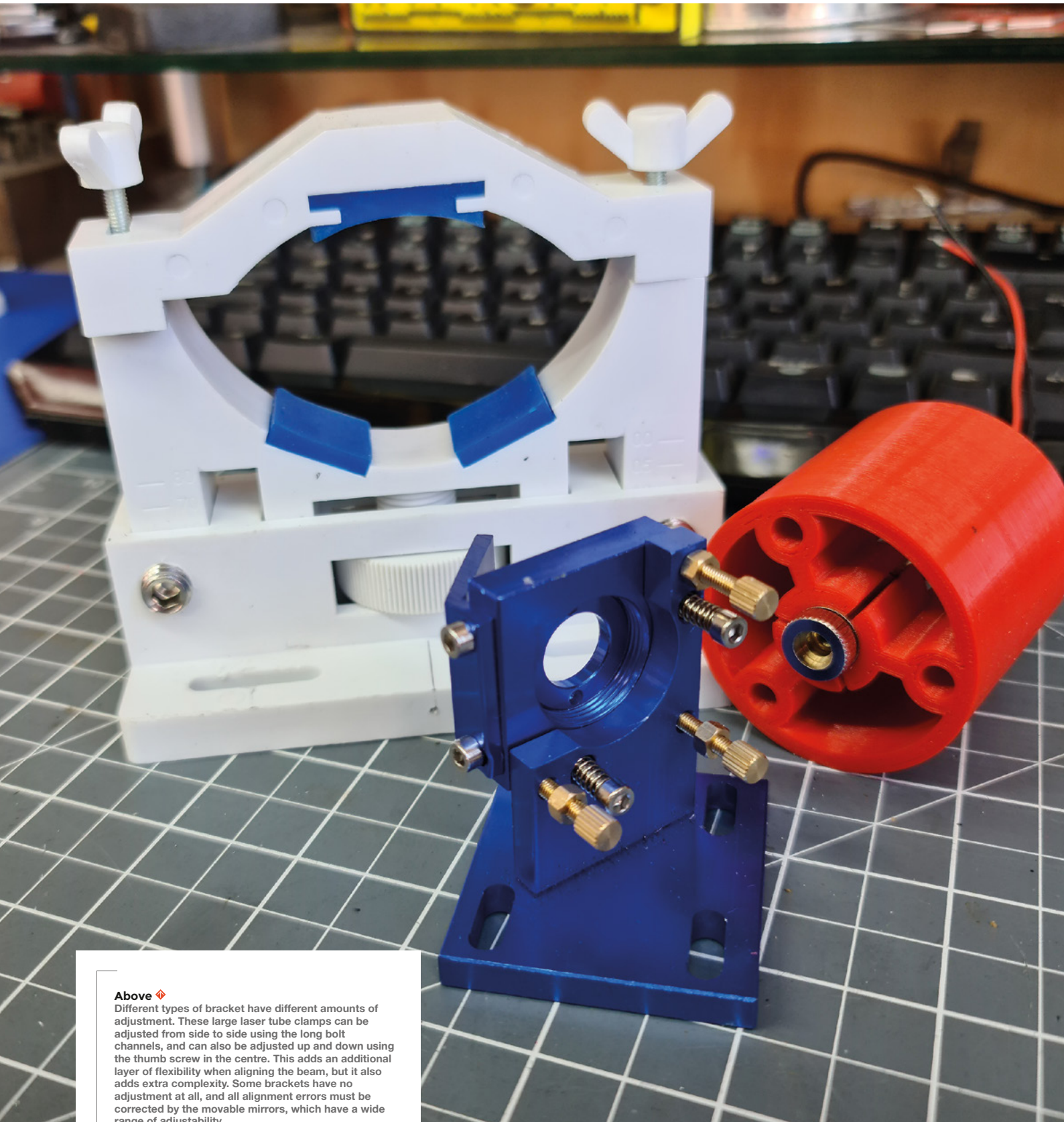


the beam is bouncing to inside the laser chamber. Place your first mirror so that the beam striking it will be reflected to the next mirror at its furthest position from the laser tube. Move on to position the next mirror in the same way, and repeat this process for each mirror. Move the gantry so that each mirror is positioned in its opposite extreme, and repeat the process until your beam is consistently visible on the bed of the machine, regardless of the position of the gantry. For the Z axis, remove the lens from the head if it is already installed, and use a piece of tape to mark the position of the laser beam projected onto the bed, then lower the machine bed. Adjust the head as necessary to get the laser point in the same place at different bed heights.

Once you have the mirrors aligned with the visible laser, you can remove it and place the CO₂ laser tube back into its clamps. In an ideal world the laser will still line up and work perfectly, but in reality you might need to do a more traditional calibration by placing some masking tape or card over the first mirror and briefly firing the laser at its lowest power setting. The burn mark will tell you whether the beam is striking the mirror in the centre or not, and then you can make any minor adjustments to the lens angle and position before tightening the bolts to lock it into place. Move the gantry to each extreme, and test the laser on a piece of scrap material. Hopefully everything should line up correctly, but if you're still having problems, →

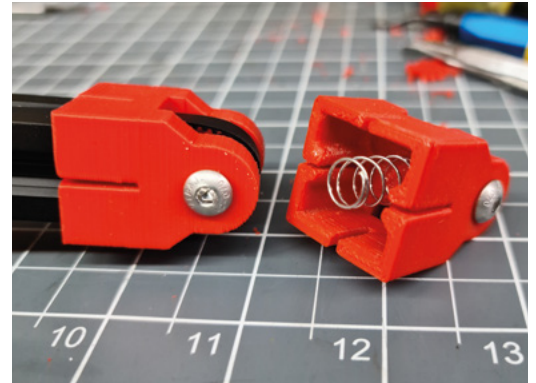
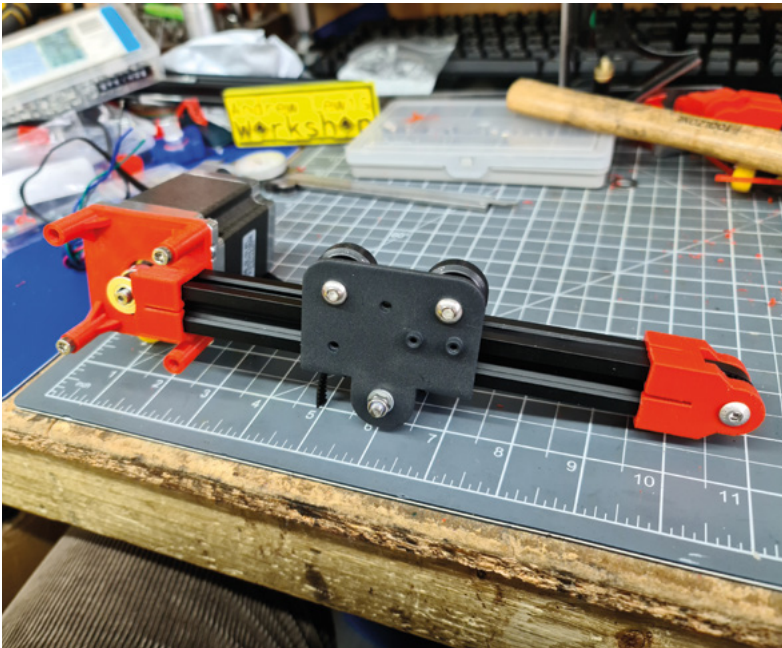
Left

A visible laser can still be dangerous, and you still need to wear appropriate laser protection when working with them. The safety goggles for visible laser light are different to those for the IR CO₂ laser, so make sure you're wearing the right one while you work



Above ♦

Different types of bracket have different amounts of adjustment. These large laser tube clamps can be adjusted from side to side using the long bolt channels, and can also be adjusted up and down using the thumb screw in the centre. This adds an additional layer of flexibility when aligning the beam, but it also adds extra complexity. Some brackets have no adjustment at all, and all alignment errors must be corrected by the movable mirrors, which have a wide range of adjustability



Left & Above

This 2020 aluminium extrusion makes it easy to attach motors, and the slots can even hide toothed belts, offering better protection than an exposed belt. Sprung idler pulleys maintain pressure on the belt to prevent it from slipping

you can repeat the same process for each mirror, using tape to see where the laser fires and then adjusting as necessary.

KEEP FIDDLING WITH IT

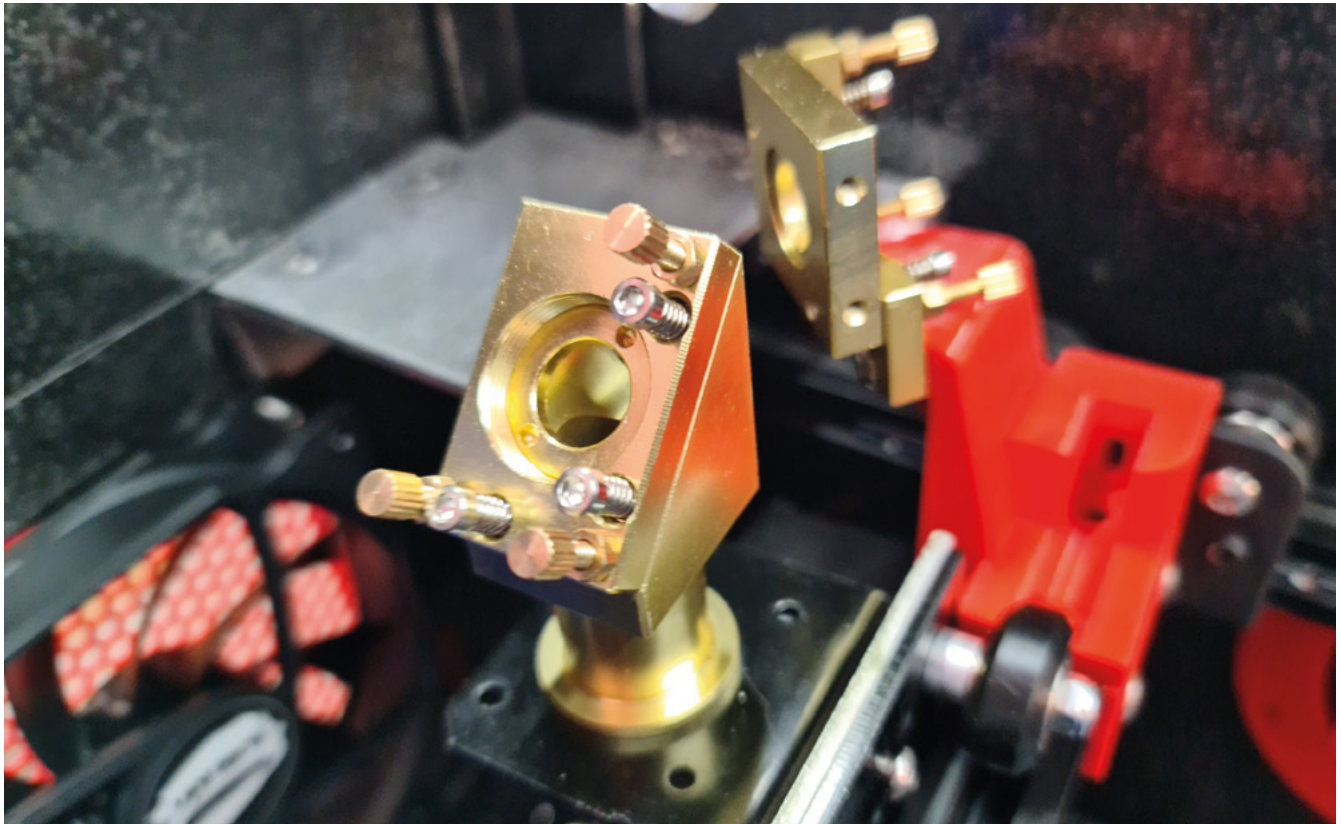
The ultimate goal is to get the laser beam running parallel to the rails of each axis so that when you're running the lens along the rail, the beam should strike the same place on the mirror all the way along the length of the rail. If it doesn't, you need to keep adjusting the alignment of the mirror until you get consistent results along the entire length. If the laser beam is hitting in a consistent position along the length of the axis, but is not close to the centre of the mirror, then you can move the whole bracket to centre it on the dot. →

UNDER TENSION

One source of errors in machines like 3D scanners and laser cutters is slippage caused by insufficient belt tension. Belt tensioners are critically important to avoid errors while cutting or printing. You have multiple options when it comes to belt tensioners, ranging from springs that clip around the belt to pull it tight, large sprung or weighted tensioner assemblies, or integrated tensioners that include the idler pulley. It doesn't really matter which style you choose, as long as you have one. Second to low belt tension, motors skipping steps can also cause problems. If the belts are too tight or bind on part of the chassis, the friction may be enough to cause the motor to skip steps. If you hear the familiar 'clack-clack' sound, check your motors are wired correctly, and your belts are running freely.



Above Actually measure the diameter of the laser tube. The number on the side of it is just a rough guide



Above & Right ♦♦

On most machines, adjustments can be made to the angle of the head by turning screws on the rear of the mirror mount, but some cheaper machines (like the K40) have a fixed head. If you have a skew on a fixed head machine, you need to figure out if the source of the skew is the mounting bracket or the rails, and correct for it accordingly. If the rail is twisted but can't be moved to a better position, you'll have to adjust the bracket with shims or washers to correct for the error



The next calibration check involves a process called **tramming**, which might be familiar to those of you who own a lathe or a milling machine. The term is often confused with **levelling**, particularly when talking about 3D printers, but **levelling** and **tramming** are two different processes. **Levelling** the bed of a machine involves making sure that it is level relative to gravity, while **tramming** makes sure that all of the important parts (like milling cutters or nozzles) are parallel or perpendicular to each other. When **tramming**, the alignment of parts in relation to each other is far more important than being parallel with the floor of the workshop. Laser cutters can be quite forgiving of small errors in their Z axis alignment, but you'll have to check for and correct any other misalignments that could cause you a problem. Too large a misalignment will push the laser out of focus and result in an uneven, shallow cut with sloped edges.

Most importantly, the laser lens assembly needs to be perpendicular to the bed of the machine. If you have any warp or twist on the lens, your laser will be cutting at an angle, and cutting through thicker materials will result in chamfered edges rather than right-angle cuts. Think of it as trying to cut through a piece of wood with the saw crooked. There are some times when you'd want to do this on purpose to form a mitred edge, but generally you want the laser to be perpendicular to the bed.

THE TRAM PROBLEM

Errors in perpendicularity between the X and Y axis will cause right angles to be cut incorrectly, turning squares into parallelograms. Testing for this is simple enough, and involves using a large engineering square with a known good right angle to check the rails of the two axes are perpendicular. How you correct the problem will vary depending on how your system is wired. If each side of your X axes has an independent motor with an endstop, then you need to adjust the endstop position and rehome the axes to get them square. If you share a single motor for both of the X axes rails or only use one endstop, you'll need to slip the pulley or the toothed belt around the shaft to better align the Y axes. Axes with a shared motor like this will need to be checked more frequently for slippage, as the controller's homing process can't correct any errors

TROUBLESHOOTING

There are a few symptoms to watch out for when assessing your laser cutter. Let's take a quick look at some of the more common ones. Poor or variable cutting power can be caused by a worn tube or dirty optics, but it can also be caused by misaligned mirrors or a lens that's been fitted into the head backwards. When using a planoconvex lens, the flat side of the lens should be closest to the object that you're cutting. The kerf of a laser beam should be tiny – less than a millimetre. If you're seeing very wide cut lines, your laser is probably out of focus.

Cut lines starting and finishing in the wrong place can be caused by the stepper motors losing steps because they are underpowered or restricted in some way, or by a slipping belt. Belt slips can also cause systems to lose the alignment between rails, resulting in skewed cuts.

Smoke staining on printed parts can be caused by excess power, or poor ventilation. Sometimes it's just a product of the material that you're working with and it can be almost impossible to eradicate completely. If smoke stains are a problem, consider masking the surface with masking tape before you start cutting, or polishing wood so that stains will wipe off easily.

with only a single endstop to work from. A similar rule applies to the Z axis of the machine, when the motors affect the tilt of the table relative to the rails. These can be set so that the optimum distance from the lens is the point where the motors find their home position. As with the lens alignment, any skew in the table can

be measured relative to the rails of the machine and adjusted with shims if necessary.

With all of the adjustments to the chassis and the mirrors complete, you can insert the laser lens and make some low-power test

dots at the corners of the gantry. Each dot should be crisp and round; if you're getting ovals or other strange shapes, you need to check your alignments again.

Even a temperature change in the workshop can be enough to push a laser reflector out of alignment, so you'll need to recalibrate your laser every few weeks, just to make sure that nothing has crept out of tolerance. Once you get the hang of it, you'll be able to get through the whole procedure in just a few minutes. □

If you're getting ovals or other strange shapes, you need to check your alignments again

Photographing the invisible

We can't see it, but we can photograph it, as Mike Bedford reveals in this hand-on guide to infrared photography



Mike Bedford

Despite loving all things digital, Mike admits to being a bit of a Luddite, vinyl records and all.

Because our eyes are sensitive to that part of the electromagnetic spectrum that we call light, and since the cone cells of our retinas respond differently to the various wavelengths of light, we're able to see the world in full colour. But beyond the long-wavelength end of the visible spectrum, that's just after red, we find an area of radiation that's known as infrared. Our eyes aren't sensitive to infrared, of course, so we might wonder how different the world would look if we could see this invisible radiation. In other words, what it would be like to mimic the visual capabilities of some insects, fish, frogs, and snakes? The good news is that, by employing a special form of photography, we really can see infrared and it's not nearly as difficult as you might imagine. Here, we're going to see how to do that using an ordinary camera and then, better still, by making your own infrared camera using a Raspberry

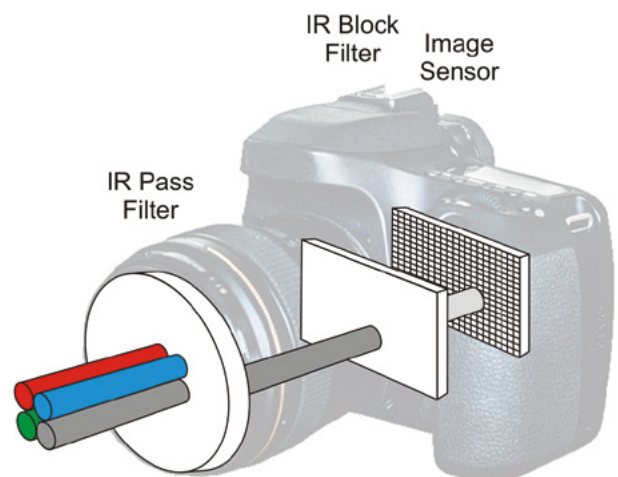
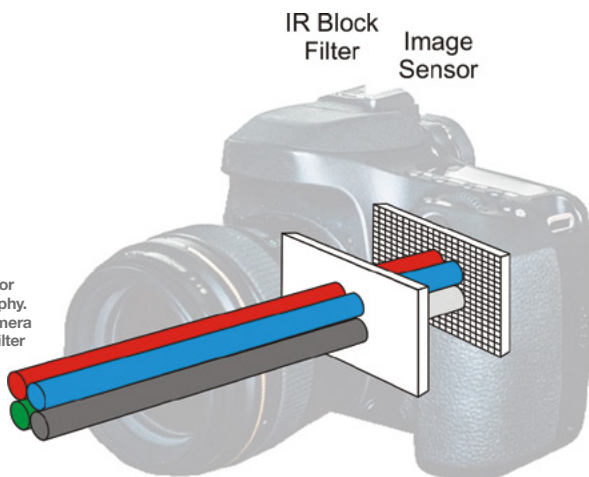
Pi and one of its associated camera modules. Not only are the results unusual but, we feel sure you'll agree, they're impressive in the extreme.

UNDERSTANDING INFRARED

Infrared, often abbreviated to IR, is often thought of as heat, but that's not true of the whole infrared spectrum. Heat is called thermal infrared, or sometimes far infrared (FIR), but that's not our subject here. Sure, you can photograph thermal infrared, but it requires specialised and expensive equipment, and it's mostly used for scientific or technical applications, not for artistic pursuit.

Instead, we're going to see how to photograph in the near infrared (NIR), which is the portion of IR just beyond the limit of human vision. The good news is that the image sensors in digital cameras are sensitive to NIR. But to create images using only IR, we need to prevent visible light getting into the camera. This is done using a filter over the camera's

Right ♦ Using an ordinary camera normally. Middle: Using an ordinary camera for infrared photography. Right: Using a camera with no IR block filter for infrared photography – this is the only solution which doesn't filter out most of the infrared





Left ♦ The classic infrared look is monochrome but with high-contrast dark skies, while grass and foliage almost seem to glow

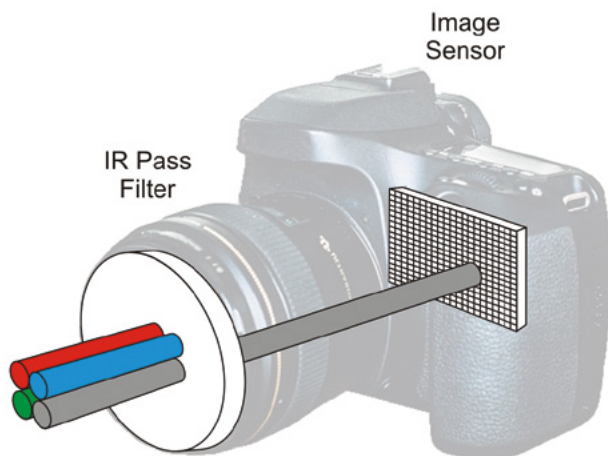
lens, that will allow IR to pass while blocking visible light – we’ll call this an IR pass filter. There’s a snag, though. Camera manufacturers use a different type of filter internally to block IR – we’re calling it an IR block filter – to prevent the unnatural colour cast that would otherwise occur. So, if you fit an IR pass filter over your camera or phone’s lens, you’ll be able to take IR shots, but your camera won’t be very sensitive. Even so, it’s a good place to start because it’s so easy. See the first two parts of the diagram below left to get a better understanding of these filters. In the first part, we see how in normal photography, the IR block filter very much reduces the amount of infrared (shown as grey) reaching the image sensor but visible red, green, and blue light are unaffected. In the second part, we see how an IR pass filter can be used to block out the visible light to take a purely IR shot, but the amount of infrared reaching the sensor is still reduced. We’ll come to the third part later.

GETTING PRACTICAL

Using an ordinary camera with an IR pass filter gives you an infrared camera which isn’t very sensitive – so you need long exposures – but don’t dismiss it. Until recently, I’d taken all my IR shots this way,

and I’ve had some good results. However, before buying an IR pass filter, do check that your camera or phone is sensitive to IR. I haven’t found one that wasn’t, although they do vary in their sensitivity, and older cameras might be better because they have less effective IR block filters. A good way to check is to observe the end of a TV remote control on your camera’s LCD viewfinder – the end you point at the TV – in a dim room while you press a button. Or, if you have a DSLR that doesn’t use the LCD panel as a viewfinder, you’ll have to actually take a photo. If your camera can see infrared, you’ll see a small bright circle at the end of the remote control unit.

If you’ve made sure your camera is IR-sensitive, it’s safe to buy an IR pass filter – just search for ‘IR filter’. A good choice is an R72 filter, which has a 720 nm cut-off, because it’ll give you shorter exposures than an 850 nm filter. If your camera has a filter thread, the filter needs to be of the same size or, if it doesn’t, like most compact cameras and phones, just buy any filter that’s big enough to cover the lens. Prices vary from about £8 to £100, but we suggest a fairly cheap filter to start. →





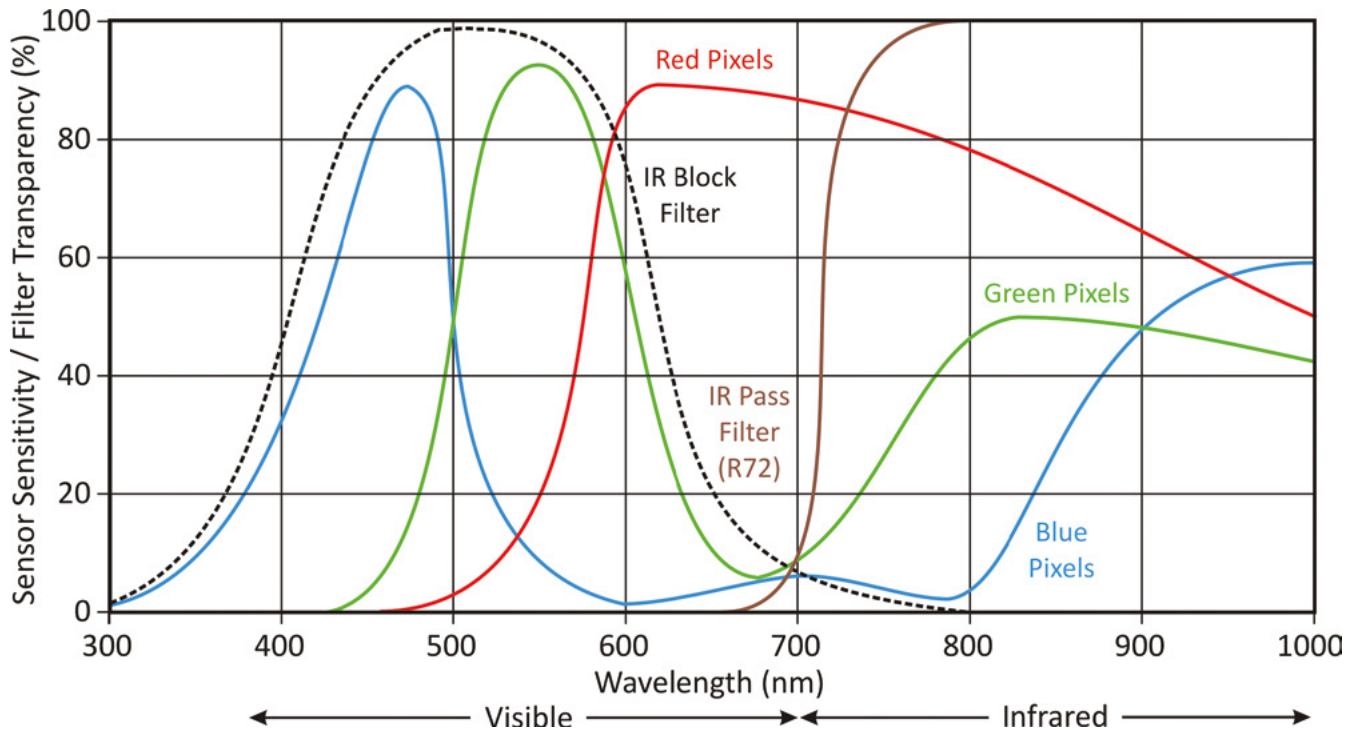
Above ♦ You could design your own Pi NoIR-based IR camera, as you can see from this one that was produced by experimenter Rob Gill, and which we used to take all the IR photos presented here. Alternatively, you'll find no shortage of ready-to-go projects

Practically, there's not a lot to say except that you should screw on, or otherwise fix, the filter over your camera's lens and you're then ready to go. However, because of the long exposure times – perhaps a second or more, even in bright sunlight – you should use a tripod. Tripods don't have to cost a lot. Alternatively, you could use some support like the top of a wall to steady your camera during the exposure. About the only other thing you need to know is what to photograph. You'll soon learn what makes a good IR photo, and the photos reproduced here will show you some of the classic effects,

namely dark blue skies which are sometimes virtually black, and vegetation that's almost glowing. To use the much-used adjectives, infrared photos have been described as bizarre, weird, surreal, eerie, dreamlike, otherworldly, and spooky. And scenes like graveyards and derelict buildings have become something of a cliché among IR photographers. Depending on your camera, IR photos might appear like black and white photos on steroids, which is what we want, but if they have a colour cast as they probably will, just convert them to true black and white using photo editing software. Perhaps also increase the contrast.

THE RASPBERRY PI ALTERNATIVE

Using an ordinary camera and an IR pass filter is a great way to get a feel for infrared photography, but most serious IR photographers are put off by the lack of sensitivity. So, they either get a camera modified by a specialist who removes the internal IR block filter, or they convert it themselves. But the former is expensive and with the latter, unless you're careful or lucky, you could end up with a load of parts you can't put back together. We have a different solution. Raspberry Pi produces a version of its V2 camera module that doesn't have an IR block filter – it's called the NoIR. It's commonly used with an IR light source for night-time security applications, but it's also suitable for artistic IR photography.



Compared to some cameras, the resolution is modest, at 3280×2464, but you could easily find that adequate, indeed we concentrated on it here because it's such a cost-effective way into blocking-filter-free IR photography. However, for unexplained reasons, we experienced blurring towards the edges of the photos. A better alternative would be based on a Raspberry Pi High Quality (HQ) Camera. It has a higher resolution, larger sensor, and larger interchangeable lenses which have adjustable aperture and focus, and is still affordable. There isn't a version without the IR block filter, but it's a fairly simple job to remove it, and there are instructions on the Raspberry Pi website, although it does invalidate the warranty. The working of a camera with no internal IR block filter, but an external IR pass filter, is shown in the third part of the diagram on page 91.

Using a NoIR or HQ Camera Module requires a bit more work than with an ordinary camera because it needs to be housed together with a Raspberry Pi board, an LCD panel, and a power supply,

so you have a portable unit. We're not going to suggest exactly how to do that, because some of the decisions are matters of personal choice, and there are lots of published designs to choose from. Most of those designs will use an ordinary Camera Module v2, not the NoIR version, or an unmodified HQ Camera, but swapping a Camera Module v2

for a NoIR camera or removing the filter from an HQ Camera barely impacts the design.

About the only addition you'll need – and this only applies to the NoIR because the HQ Camera lenses have filter threads – is some

means of attaching the IR pass filter over the lens. If you don't need to use the filter elsewhere, you could glue it in place; otherwise, glue on a filter stepper ring so that you can screw the filter over the lens.

INFRARED COLOUR

Conventionally, infrared photos are monochrome, but there are several ways of creating colour IR shots. The methods involve post-processing and you'll need a fully featured editor but, if you don't →

Above ♦ Just as red, green, and blue pixels have different peaks in the visible spectrum, they do in the infrared too. This means that, if your camera has its IR block filter removed, you can capture false colour information in the infrared. These graphs are typical, but will differ between cameras

Using a NoIR or HQ Camera Module requires a bit more work than with an ordinary camera



“

Remember, though, that we're processing the invisible so there's no correct way for these photos to look

”



have one, we suggest GIMP, which is free. We don't have space to provide step-by-step instructions, but we'll give enough guidance for you to figure it out yourself.

With an ordinary camera, i.e. one with an internal IR block filter, the methods involve taking two photos, an ordinary one and one taken with an IR pass filter over the lens, and combining them. One of the best ways is to split the normal photo into hue, lightness, and saturation channels, discard the lightness channel and replace it with the monochrome infrared photo before recombining them to make a colour photo. The colours will be normal (-ish) but with the classic features of an infrared photo, like dark skies and bright foliage.

The second method – which we call false colour – requires only an infrared photo, but it should have been taken on a camera with no IR block filter. The graph on page 93 helps you to understand how this works. It shows how a typical camera, not any particular one, responds to different wavelengths. We can see that the camera has peaks for red, green, and blue in the visible part of the spectrum, as we'd expect, but the camera's red, green, and blue pixels also react differently to different parts of the IR spectrum. In other words, there is 'colour' – although we don't use that word in its strictest sense – in the infrared. If the camera has an IR block



filter, though, the camera will only be sensitive to a small part of the IR spectrum, between where the IR pass filter (e.g. R72) starts and where the IR block filter ends. In our graph, that's from about 675nm to 780nm and, since the red, green, and blue pixel sensitivities don't change a lot in that range, you can see why it only works well if the camera has no IR block filter.

Straight out of the camera, images will probably have a strong red or magenta colour cast, after all, as we can see from the graph, the red pixels are the most sensitive to IR. However, the actual colour depends on the setting for white balance and whether you shoot in JPEG or RAW, so you'll

have to experiment. So, in post-processing, define a white point from an area of the photo that ought to be white – for example, clouds or grass. This should remove the red cast by boosting the green and blue pixels. Because the sky will be red or brown rather than blue, though, it's common to swap the red and blue channels. The sky will now be blue and, while foliage will be very light like IR monochrome photos, it might have a pale orange, yellow, or pink hue which many people think is a good look. Remember, though, that we're processing the invisible so there's no correct way for these photos to look, so feel free to experiment to your heart's content, perhaps increasing the saturation and adjusting the hue. ▢

Above  We prefer the false colour effect that renders grass and foliage in a pale peach or apricot colour, but the effect with almost white vegetation is quite pleasing too, and no, that's not snow on the ground

Make a printing party camera

Create a camera that sends pictures to a portable Bluetooth printer.
Use it at parties (or anywhere else) to capture the fun



Rob Miles

🐦 @robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at robmiles.com and follow him on Twitter at [@robmiles](https://twitter.com/robmiles)

These days, no big social occasion is complete without a photo booth where guests can take instant pictures of themselves. Now, you can make your own device to do the same thing. It will detect when subjects arrive and offer them the chance to immortalise themselves on 57 mm-wide thermal paper with the touch of a button. The camera and the printer are connected wirelessly over Bluetooth, and both devices are battery-powered. So, you can turn any event into a picture party. The picture quality is not the best, but it does provide a recognisable and unique memory.



Right ♦
The camera can be attached to a tripod using the built-in tripod bush

The printer is a battery-powered device that provides a Bluetooth endpoint that can accept image printing commands. There are lots of low-cost 'receipt' and personal printers available. The 'MobilePrinter' that the author used was purchased for around £15 (\$18), and works well enough.

The camera can be powered via the micro USB connector on the TTGO ESP32 module. You can add an internal battery which should be large enough to keep it going for a few hours. There is no power switch. The camera switches on when the passive infrared (PIR) sensor detects a person. When the camera awakens, it displays a start sequence and invites the user to press the button on the front to take a picture. Pressing the button starts a five-second countdown, after which a picture is taken and printed. The camera automatically powers down after 30 seconds of inactivity. The OLED screen displays large friendly messages to keep the user informed of what is going on. You can find all the box design, parts list, and all the software on the GitHub site for this project here: hsmag.cc/PartyCamera.

INSIDE THE BOX

All the electronics for the project are on a single device – a TTGO ESP32 Camera Module. You can see the back of this in the bottom right of **Figure 1**. At the top right of the box, you can see the back of the 3/8-to-1/4-inch tripod socket adapter which is screwed into a hole in the box and exposes a 1/4-inch standard tripod socket on the outside of the box. The shutter button on the bottom left is connected across the GND and IO22 connections on the TTGO expansion connector which you can see on the right of the board in **Figure 1**. The first version of the device used wire-wrapped connections to the pins in this connector.

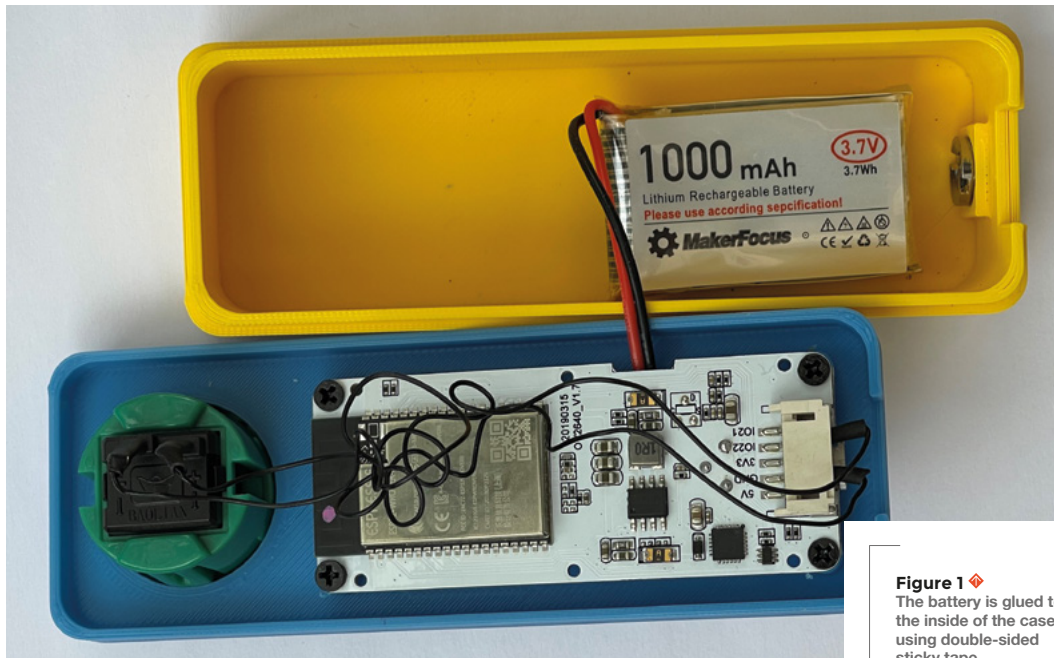


Figure 1 ♦ The battery is glued to the inside of the case using double-sided sticky tape

You could instead solder wires to the back of the connector, or you can buy a cable (search for 'LilyGO TTGO camera cable') which plugs into the connector.

Figure 2, overleaf, shows the front of the TTGO module. On the left of the board is the micro USB connector, next along is a PIR sensor, followed by the camera and 128×64 OLED display. There are also two push-buttons: one is connected to the processor reset and the other to General Purpose Input/Output (GPIO) pin 34. The GPIO button could be used as a shutter button for the camera, but it was decided that a large arcade button would be more fun and easier to press.

STARTING UP THE CAMERA SOFTWARE

The camera is controlled by a program written in C++ using the Arduino software environment inside a PlatformIO project. The code contains a **setup** function to start the device and a **loop** function which is called repeatedly when the device is active.

```
void setup()
{
  Serial.begin(115200);
  setup_shutter();
  setup_display();
  setup_pir();
}
```

The above code shows the beginning of the **setup** function which sets up the shutter input, OLED display, and PIR sensor. Setting up the shutter and

the display is quite simple, but setting up the PIR sensor is more interesting:

```
#define PIR_PIN 33

void setup_pir()
{
  pinMode(PIR_PIN, INPUT);
  esp_sleep_enable_ext0_wakeup((gpio_num_t)PIR_PIN, HIGH);
}
```

The code above shows the function **setup_pir** that sets up the PIR sensor. The symbol **PIR_PIN** specifies the GPIO to which the sensor is connected. On the TTGO board, the PIR sensor is connected to GPIO33. The first statement in **setup_pir** sets this pin to be an input pin. The second statement uses the **esp_sleep_enable_ext0_wakeup** function to tell the ESP32 to wake up and start running when the PIR sensor lifts **PIR_PIN** high. The next thing the **setup** function does is make sure there is someone there to photograph.

```
if (!digitalRead(PIR_PIN)) {
  go_to_sleep();
}
```

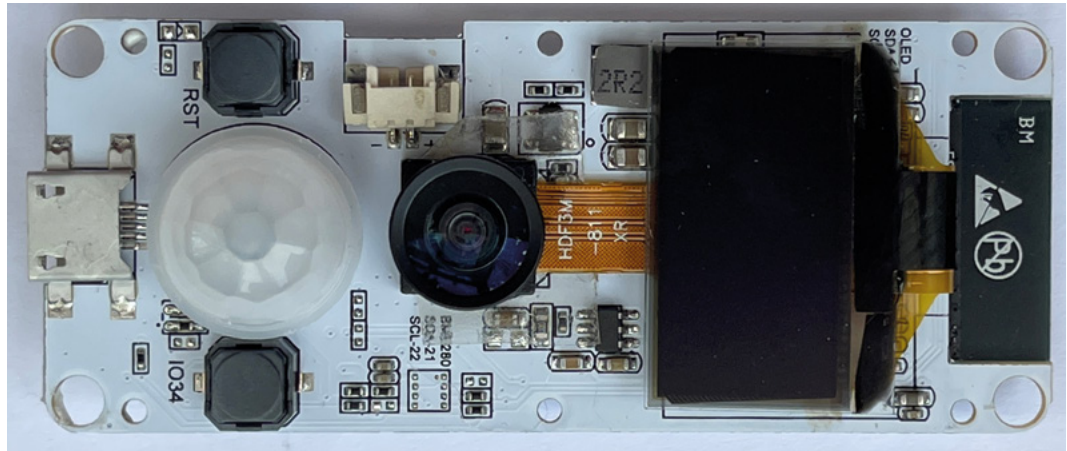
The code above tests the PIR input. If the input is low, this means that there is nobody in front of the camera so it should shut down. →

YOU'LL NEED

- ♦ **Bluetooth thermal printer**
Search for 'Mini Portable Bluetooth Thermal Printer'. They are usually supplied with a small roll of 57 mm-wide thermal paper. You can get more from any office supply shop
- ♦ **ESP32 camera board.** The 'TTGO ESP32 camera' unit is perfect for this project. It has a display and a PIR sensor to wake up the camera. Search for it by name
- ♦ **Battery.** The TTGO device can connect to a rechargeable battery. Search for '1000mAh 802540 3.7 volt' if you want one that works with the case design
- ♦ **A button to activate the camera.** Search for '24mm arcade button'. The author used wire-wrap wire to connect the button to the TTGO. You can also get a connector that you can use rather than wire wrap. Search for 'TTGO camera cable'
- ♦ **A box.** There is a 3D-printable design available. If you want to put the camera on a tripod, you will need a 3/8" to 1/4" camera adapter which you can search for by name
- ♦ **Screws.** You'll need some screws size M2 4 mm in length to fix the TTGO device to the case (search for 'laptop screws')
- ♦ **Micro USB cable.** You program and power the camera via the micro USB port on the TTGO module

Figure 2 ♦

The TTGO board can be bought with one of two different camera lenses. This board is fitted with the 'fish-eye' which gives a very wide field of view. For pictures of individuals, the 'normal' version works better



```
void go_to_sleep()
{
    // turn off the screen
    u8x8.setPowerSave(true);
    // disconnect the printer
    tpDisconnect();
    // turn off bluetooth
    btStop();
    // enter deep sleep
    esp_deep_sleep_start();
}
```

The code above shows the `go_to_sleep` function. This turns off all the devices on the camera and then puts the ESP32 to sleep. If the PIR sensor is triggered, the ESP32 will reset and the camera software will start running again and call `setup`.

WHEN IN DOUBT, SLOW THINGS DOWN

First attempts to use the camera on the TTGO board were not encouraging. The camera was not always detected. There are a variety of possible reasons for failures like this. Perhaps the power supply glitches when the camera powers up. Perhaps the reset pulse the ESP32 sends to the camera is too short. Perhaps the connections between the devices are noisy. Perhaps the camera is not responding quickly enough to requests from the ESP32. In the case of the TTGO camera, it was the last issue. The fix that worked was to slow down the ESP32 processor so that the camera could keep up. This is very easy to do. If you are using the Arduino IDE to develop your program, you can find the setting in the Tools menu there.

If you are using PlatformIO to write your code, you can add this line to the `platformio.ini` file for your project:

```
board_build.f_cpu = 160000000L
```

Slowing down the processor in a device will reduce the power consumption, which can be a good thing if you are worried about battery life. However, if you slow the CPU speed down below 80MHz, Bluetooth and WiFi will stop working.

DEALING WITH FAILURE

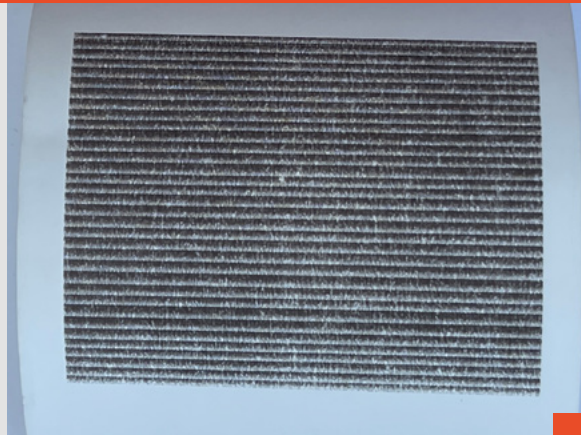
Hardware designers must deal with situations where parts of a system might not work every time. During testing, it was noticed that sometimes the camera module on the TTGO board was not detected. Furthermore, attempting to connect to a Bluetooth printer is not always successful. The printer may be switched off or connected to another device. Normally, a user could attempt to fix these problems by turning the device off and on again to force a retry. Unfortunately, the camera doesn't have an on/off switch, so we must make our code able to deal with failures.

```
bool try_setup(bool (*try_setup_function)(),
               int retry_gap_millis,
               int no_of_retries)
{
    while ((no_of_retries-- > 0) {
        if (try_setup_function())
            return true;
        delay(retry_gap_millis);
    }
    return false;
}
```

The code above shows how the camera software sets up devices which might not work first time. The `try_setup` function repeatedly calls a `setup` function. It accepts three arguments. The first is a pointer to the `setup` function to use. The second is the interval between retries in milliseconds, and the third argument is the number of retries. The body of `try_setup` uses a `while` loop to repeatedly call the supplied `try_setup_function` until the function succeeds, or it fails after the specified number of

PICTURE QUALITY

A thermal printer works by heating up elements in the print head which cause chemicals on the printer paper to turn black. When printing text, the amount of 'black' that needs to be printed at any time is quite small. However, when printing images, there may be quite a lot of black to print, and not all printers handle this equally well. The image to the right shows the result of trying to print a completely black image on the thermal printer used for this project. You can see how the print head has 'given up' towards the end of some of the rows. The author is quite happy with this level of quality for the price, but you might be able to find printers that can produce better results.



Left ♦ The image should be completely black. The bands are caused by the print head overheating, or by the limitations in the power supply

attempts. If the setup works, the function returns true, otherwise, it returns false.

```
if (!try_setup(try_setup_camera, 200, 5)){
    esp_restart();
}
```

The code sample above shows how **try_setup** is used to call the **try_setup_camera** function five times, with a 200-millisecond delay between each attempt. If a call of **try_setup** returns true (indicating that the camera has been set up), the program continues running. If a call of **try_setup** returns false, the function **esp_restart** is called to reset the device completely. The program also uses **try_setup** to connect to the printer. We have already seen that each time the device resets, it will check to see if there is anyone in front of the camera before continuing. This means that if the printer is not powered on, the camera will keep trying to connect until it detects that nobody is trying to take a photograph, at which point it will shut down.

PRINTING OVER BLUETOOTH

A Bluetooth device exposes services, each of which contains several characteristics. A client connects to a service and can send messages to the service by writing values into a characteristic which are then picked up by code in the service. Services and characteristics are identified by large numbers. A particular value means "this is a printer service" and another value means "this is a characteristic that accepts text to be printed". In addition, each Bluetooth device has a specific ID which is a string of text. The thermal printer used for this project has the Bluetooth ID 'MPT-II'. You can connect to this

from another Bluetooth device, find the printing service, and then send text to it.

The screenshot in **Figure 3**, overleaf, shows the text 'hello world' being sent from a mobile phone into the MPT-II printer using a program called 'BLE Scanner' from Blue Pixel Technologies (bluepixeltech.com). This is an excellent way to discover the services and characteristics of a Bluetooth device. You can see parts of the very large numbers that are used to identify services and characteristics. We could write our own code to drive the printer, but it is much easier to use a library written by Larry Bank, which you can find here: hsmag.cc/ThermalPrinter. This finds and connects to Bluetooth printers and works a treat. The only change the author needed to make to the library, to make it work with his printer, was to add the printer's name (MPT-II) to the list of printers the library supports. There are instructions on how to do this on the GitHub site for this project.

```
bool try_setup_bluetooth_printer()
{
    if (tpScan()){
        if (tpConnect()){
            tpSetWriteMode(MODE_WITHOUT_RESPONSE);
            return true;
        }
        else{
            return false;
        }
    }
    else{
        return false;
    }
}
```

QUICK TIP

The camera unit on the TTGO module just 'dangles' on the ribbon cable, which can make for wobbly pictures. Use a small piece of double-sided sticky tape to hold it in place on the TTGO board.

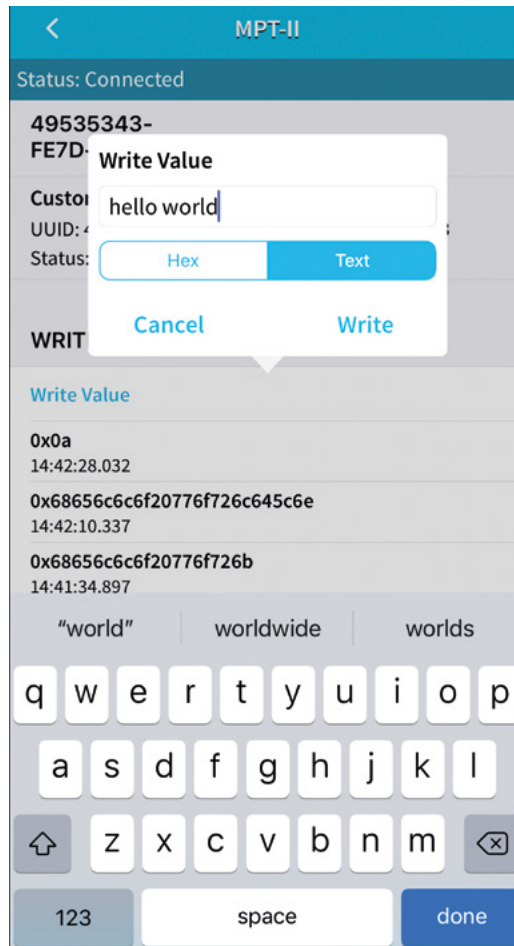


Figure 3 ♦ You need to send a message containing the hex value '0a' (line feed) to get the printer to print text you have sent it

The preceding code shows the `try_setup_bluetooth_printer` function which connects to the printer. The `tpScan`, `tpConnect`, and `tpSetWriteMode` functions are part of the thermal printer library.

Once the camera has been set up, the `update` function is repeatedly called to operate the camera. Let's look at that next.

PHOTO LOOPY

```
void loop()
{
  update_timeout();
  update_shutter_button();
  delay(10);
}
```

The code above shows the `loop` function for the camera. First, the `loop` function calls `update_timeout` to check if the camera should shut down.

```
void update_timeout()
{
```

```
  unsigned long timer_millis = millis();
  if (digitalRead(PIR_PIN)) {
    // reset the timer if people are still in
    front of the camera
    start_time = timer_millis;
  }

  if ((timer_millis - start_time) > TIMEOUT_
  MILLISECS) {
    // turn off the camera
    go_to_sleep();
  }
}
```

This function turns off the device if nobody has been detected for a while. It also resets the timer if there is someone in front of it so that the camera stays awake when it is being used. The `go_to_sleep` function is called to put the camera to sleep. We have already seen how this function is used when the camera starts up and there is nobody using it.

TAKE A PICTURE

The second statement in the `loop` function calls the `update_shutter_button` function to check if a picture should be taken.

```
void update_shutter_button()
{
  bool new_shutter = digitalRead(SHUTTER_PIN);
  if (last_shutter_input != new_shutter) {
    if (new_shutter == 0) {
      take_picture_and_print();
    }
    last_shutter_input = new_shutter;
  }
}
```

The code above is for the `update_shutter_button` function. This detects when the button has been pressed down by comparing the new shutter input (`new_shutter`) with the last one (`last_shutter_input`). If the two are different, the button has changed state. If the new state of the button is 0, this means that the button has been pressed down, and the camera must take a picture and print it. This action is performed by the `take_picture_and_print` function. This function displays a smile countdown and then calls the functions `take_picture` and `print_picture` to produce the picture output.

```
uint8_t *frame_buffer = NULL;
int frame_buffer_size;
```

The preceding code shows the variables in the program that hold information about an image. The `frame_buffer` variable refers to a buffer of pixel data and `frame_buffer_size` specifies the number of pixels in the buffer. These variables are global. The `take_picture` function puts values into them and the `print_picture` function reads them.

```
void take_picture()
{
    frame_buffer = (uint8_t *)cam.getfb();
    frame_buffer_size = cam.getSize();
}
```

The code above shows the `take_picture` function which sets the frame buffer and frame buffer size by activating the camera.

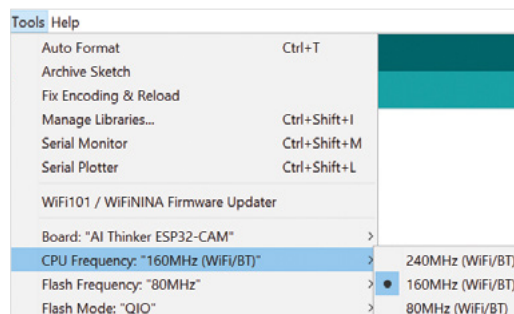
```
void print_picture()
{
    if (jpg.openRAM(frame_buffer, frame_buffer_size, printRows))
    {
        jpg.setPixelType(ONE_BIT_DITHERED);
        jpg.decodeDither(ucDither, JPEG_SCALE_HALF);
    }
}
```

The `print_picture` function shown above prints the image stored in the frame buffer. It uses another library written by Larry Bank. The JPEGDEC (hsmag.cc/JPEGDEC) is used to decode the image from the camera and send it to the printer. The `print_picture` function sets up the JPEG decoder and then uses the `decodeDither` method from JPEGDEC to decode the image from the camera. As the image is decoded, `decodeDither` calls the `printRows` function to send rows of image data to the printer.

```
int printRows(JPEGDRAW *pDraw)
{
    int i, iCount;
    uint8_t *s = (uint8_t *)pDraw->pPixels;

    tpSetBackBuffer((uint8_t *)pDraw->pPixels,
                    pDraw->iWidth, pDraw->iHeight);
    iCount = (pDraw->iWidth * pDraw->iHeight) / 8;
    for (i = 0; i < iCount; i++){
        s[i] = ~s[i]; // find inverted patterns in s
    }

    tpPrintBuffer(); // Print this block of pixels
}
```



```
return 1; // Continue decode
}
```

The code above shows the `printRows` function, which sends image bytes generated by `decodeDither` to the printer. Each byte in the image data gives the state of eight dots across the printed row. These need to be inverted because the camera image uses a set bit to indicate white, whereas the printer uses a set bit to indicate that a black dot is to be printed. The buffer is assembled and then printed by the `tpPrintBuffer` function.

GOING FURTHER

There is a lot of scope for extending this project. You can use the printer to add bar-codes, QR codes, and text to an image. You could create a ticket creator application which has a picture of the ticket holder and a QR code holding the URL for the event. Or a simple photo ID. The author hopes that you have learned some tips about creating devices that power-up automatically, capturing images and using a Bluetooth printer from an ESP32. And just maybe, if you make the party camera, you might get invited to even more parties. ■

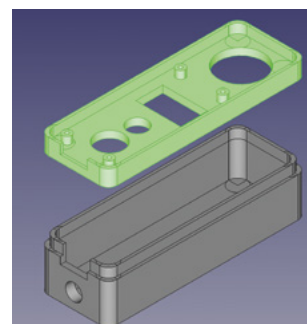
PRINTING TINY TEST PIECES

It is often a good idea to print parts of a design to make sure that they fit before you print the entire thing. Below, you can see a test piece that was created to make sure that the opening in the camera base was the right size for the USB plug. It is very easy to use a CAD tool to cut large chunks away from your design so that you can print just small test parts to make sure that they fit.



Left ◆
The AI Thinker ESP32-CAM board is a good match for the TTGO device

Below ◆
You can screw a 3/8 tripod adapter into the bottom of the case so that the camera can be attached to a tripod



HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated



PG
112

WAVESHARE LORA

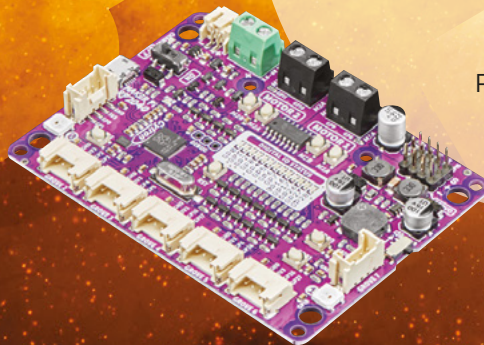
Adding distance to your
wireless communications



PG
104

BEST OF BREED

Our pick of
Raspberry Pi robots



DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE
FOR JUST
£10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door

+ FREE
RASPBERRY PI
PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.



magpi.cc/subscribe

ONLY THE
BEST

Raspberry Pi Bots & Rovers

Get your Raspberry Pi moving

By Marc de Vinck

 @devinck

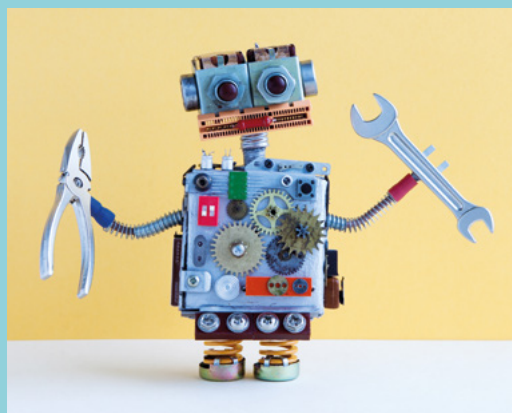
The Raspberry Pi made waves when it entered the market, and for good reasons – it was affordable, powerful, and relatively easy to use. We have come a long way since the introduction of the original Raspberry Pi back in 2012.

Now you have a choice of small form factor Raspberry Pis such as Raspberry Pi Zero W and the incredible power you can unleash from Raspberry Pi 4.

But there is one issue plaguing the Raspberry Pi community, and that's availability. Although I can't say exactly how many Raspberry Pi boards have been sold, I can say it's a lot! But as of lately, they have become difficult, but not impossible, to find in stock. And it's not just Raspberry Pis that have become hard to find, it's basically everything, especially electronics. And it stinks!

But there are a lot of DIY-ers out there that have an extra Raspberry Pi in their stash of electrical components. You might have torn apart an old project, or you had the foresight to order a few extra when they were easier to find. Remember those days? Yeah, me neither.

In this Best of Breed, I wanted to look at some interesting robotics kits and accessories that can help you put that extra Raspberry Pi to use. Some of these kits have Raspberry Pis integrated, but most just need you to add one of your own to make it complete. And for those of you that are Raspberry Pi-less right now, keep your eye out for inventory online. I do see them pop up from time to time, and at reasonable prices.



FRIENDLY TIP

Try rpilocator.com to help you find a Raspberry Pi online!

MonsterBorg vs Trilobot Base Kit

PIBORG ♦ \$257 | piborg.org

PIMORONI ♦ \$54.12 | pimoroni.com

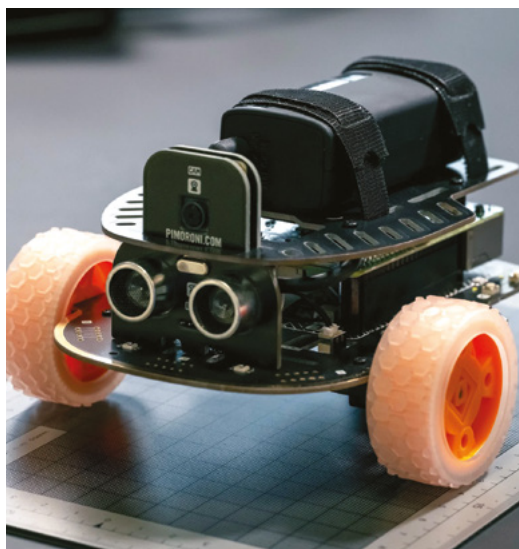
The **MonsterBorg from PiBorg is a robust robotics platform for your Raspberry Pi**. It features a custom motor driver: the ThunderBorg. This allows you to easily control a variety of DC motors directly from your Raspberry Pi. The kit includes four high-torque 300 RPM metal geared motors, a well-designed aluminium chassis, 105mm off-road tires, and a mount for a Camera Module.

No soldering is required. All you need to do is add your Raspberry Pi, almost any standard variant will do,

and you are ready to tackle the outdoors. PiBorg claims you will get up to three hours of run time on ten AA batteries. Adding a camera, GPS, and some additional sensors could make for a great autonomous outdoor robotics platform capable of withstanding a fair amount of abuse.



Above ■
For when the going gets tough



The **Trilobot from Pimoroni is a sturdy Raspberry Pi-based robotics kit**. It was designed with educational use in mind, so it also has great documentation and example files. The Trilobot features 2WD, breakout headers for additional sensors and accessories, six-zone under-bot lighting, status LEDs, and a front-facing ultrasonic sensor for measuring distance.

The PCB is labelled for ease of use, even for a beginner. Pimoroni has a full-featured Python library to help you get up and running quickly. You can pick up the Trilobot with or without a Raspberry Pi, so if you are lucky enough to have an extra one and want to build a bot, this is a great choice. Just remember, if you go for the version without a Raspberry Pi, you'll also need a battery and a USB card for your Raspberry Pi to get going. →

Left ■
We love the clear wheels

VERDICT

MonsterBorg

A very robust robotics platform.

9/10

Trilobot Base Kit

Add a Raspberry Pi and get going!

9/10

Robotics Superset with pi-top [4]

PI-TOP ♦ \$399.85 | pi-top.com

The pi-top [4] robotics kit includes 50+ metal build plates, two servo motors, two encoder motors, wheels, camera, ultrasonic sensor, and the pi-top controller. It's designed for autonomous and AI-based robotic projects. You can pan and tilt the front-facing camera ultrasonic sensor, allowing for a full 180° view of the surrounding environment.

At the heart of the robotics kit is the pi-top [4]. This module can be purchased with or without a Raspberry Pi, and the robotics kit can be purchased with or without the pi-top [4] module. The module allows your Raspberry Pi to run without wires and integrates a five-hour internal battery. It also has a heatsink, fan, UPS power circuitry, touchscreen, speaker, and easy access to Raspberry Pi's GPIO pins. It's a very professional-looking and well-made enclosure. Head over to the website to download additional resources and learn more about the different pi-top robotics kits that are available.



VERDICT

Robotics
Superset with
pi-top [4]

Well-developed
hardware.

8/10



Left ■
Add some
intelligence
to your robot

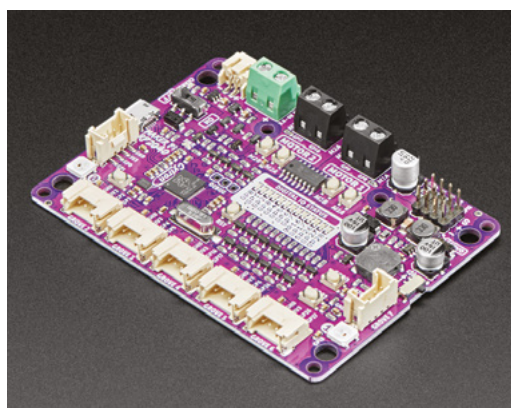
Maker Pi RP2040 – Motor and Robot Controller

ADAFRUIT ♦ \$12.50 | adafruit.com

This one is a little different from the other reviews in this Best of Breed. No, not because it isn't a full robot, but because it's not the typical Raspberry Pi. This motor controller features the relatively recently released RP2040 microcontroller, not a Raspberry Pi-based single-board computer. Either way, it still makes for a great central brain for your next robot build.

The Maker Pi RP2040 features a dual-channel DC motor driver, four servo motor ports, and seven Grove I/O connectors. It can control two brushed DC motors or one stepper motor, providing up to 1 A current per channel. It also has an array of LEDs, so you can quickly check the status of various digital I/O states. It also has an on-board piezo buzzer and several push-buttons.

To get you up and running quickly, the board already has CircuitPython preloaded and preprogrammed with a simple demo program. Just plug it into your



Left ♦
You can also use a Raspberry Pi Pico W

computer and it will immediately start playing a friendly tune. Add a few motors, click some buttons, and you're able to test your setup before you build a bot. This looks like a great way to get started with the RP2040, and best of all, they are typically in stock! →

VERDICT

Maker Pi RP2040 – Motor and Robot Controller

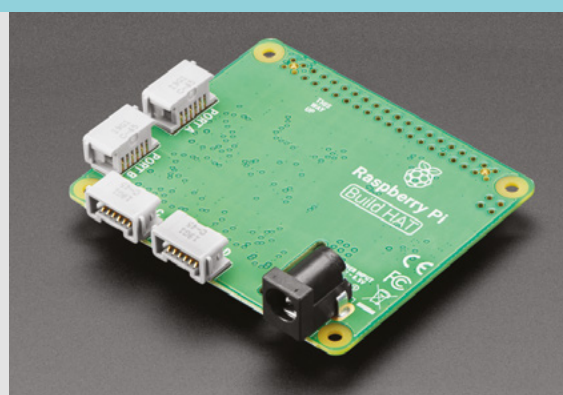
Makes building bots easy!

9 /10

RASPBERRY PI BUILD HAT

ADAFRUIT ♦ \$25 | adafruit.com

Who doesn't love LEGO? It's hard to find someone who wasn't obsessed with LEGO at some point in their life. Young or old, it's a universally fun product. And it's great to see the Raspberry Pi Build HAT allows you to integrate your Raspberry Pi and LEGO Technic motors and devices. And since all the components are on the bottom of the board, you can easily add a small breadboard or LEGO board on top of it.



RockyBorg

PIBORG ♦ \$121 | piborg.org

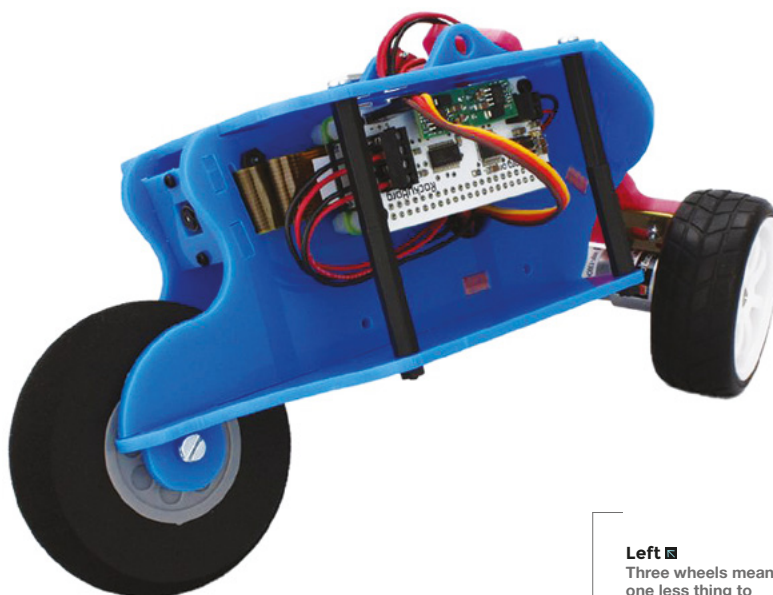
The RockyBorg by PiBorg features two 20mm 180 RPM metal geared motors, and a metal gear servo for leaning the chassis into turns. It also comes with a newly designed motor controller that is compatible with your Raspberry Pi. The body is made from laser-cut acrylic and features a unique leaning motion that would be interesting to see from the viewpoint of a forward-facing on-board camera. The website has more information, including builds for controlling the bot with a PS3/PS4 controller via Bluetooth. □

VERDICT

RockyBorg

An interesting bot platform.

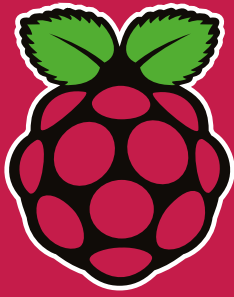
7 /10



Left □
Three wheels means one less thing to worry about



THE *Official*

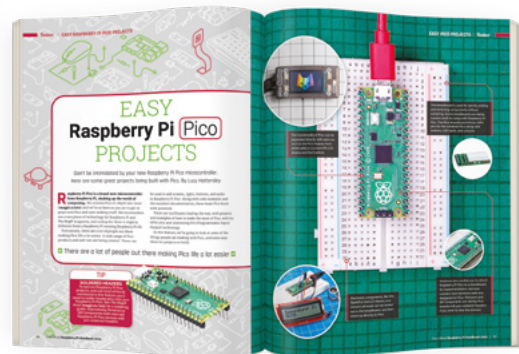


RASPBERRY PI HANDBOOK

2022

200 PAGES OF RASPBERRY PI

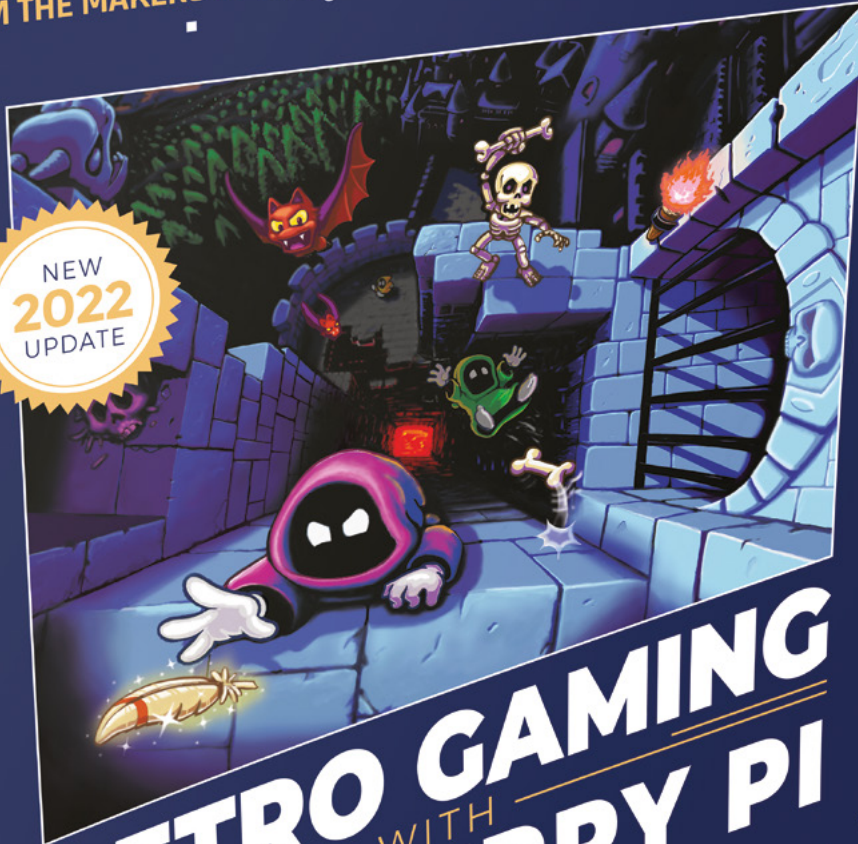
- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects



Buy online: magpi.cc/store

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

NEW
2022
UPDATE



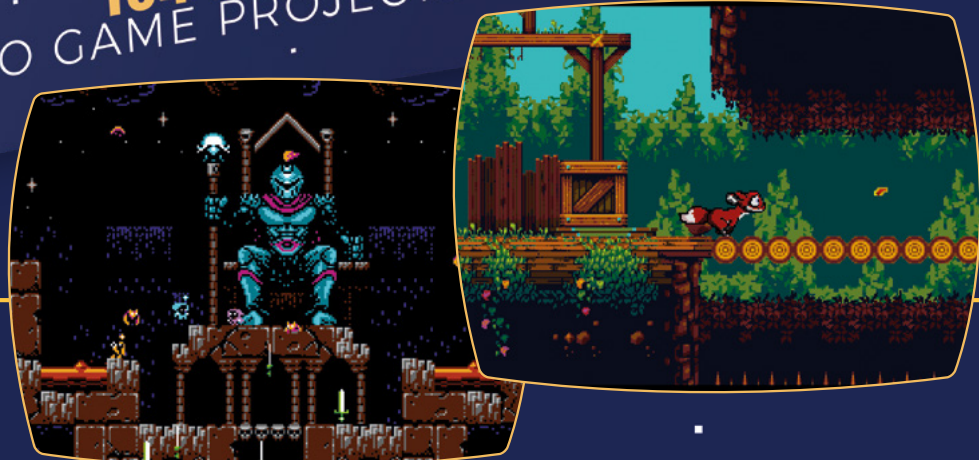
RETRO GAMING WITH RASPBERRY PI

2ND EDITION

164 PAGES OF
VIDEO GAME PROJECTS



**PLAY
& CODE
GAMES!**



RETRO GAMING

WITH

RASPBERRY PI

2ND EDITION

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own games console or full-size arcade cabinet, install emulation software and download classic arcade games with our step-by-step guides. Want to make games? Learn how to code your own with Python and Pygame Zero.

- ***Set up Raspberry Pi for retro gaming***
- *Emulate classic computers and consoles*
- ***Learn to code your own retro-style games***
- *Build a console, handheld, and full-size arcade machine*



BUY ONLINE: magpi.cc/store

Waveshare SX1262 LoRa

Add to a Raspberry Pi Pico to create a networked sensor

Waveshare ♦ £14.50 inc VAT | waveshare.com

By Rob Miles

🐦 @robmiles

Making a connected device just got a bit easier and cheaper with the launch of a module that allows a Raspberry Pi Pico to use the LoRa network.

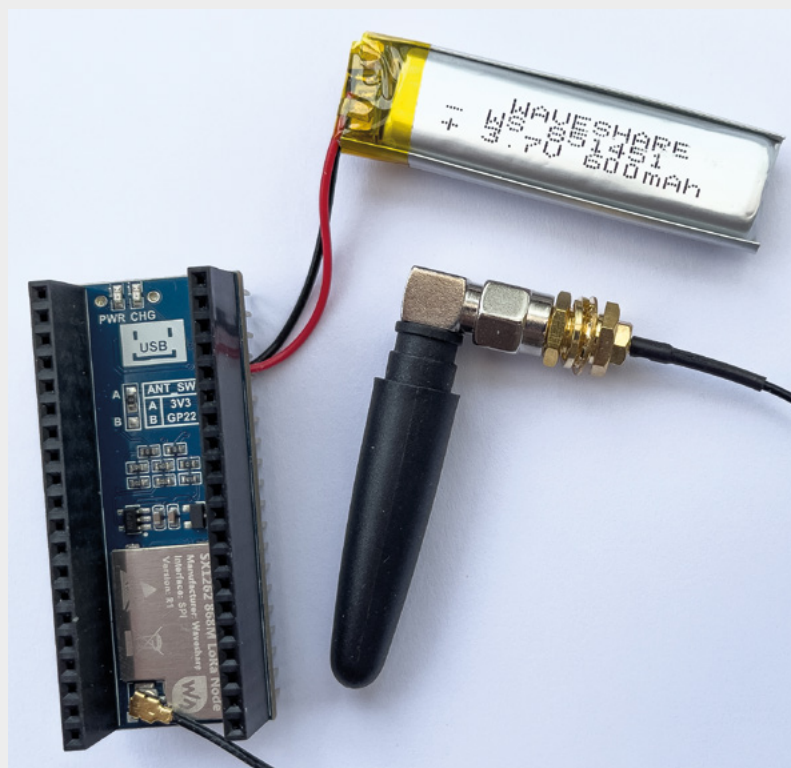
LoRa – short for ‘long range’ – is a great way to send data over considerable distances whilst operating from tiny batteries due its low power requirements.

You can use LoRa to pass small messages between devices, but it is much more useful to

make a device part of a LoRaWAN application. In an application, messages are sent to a local gateway which provides a connection to a LoRaWAN service running in the cloud. You can set up your own LoRaWAN infrastructure, but most people use ‘The Things Network’ (thethingsnetwork.org), which is free to access and underpins a huge number of community-operated gateways. You can check the gateway coverage in your area by using the TTN mapper site (ttnmapper.org). Your programs can interact with a LoRaWAN application in a variety of ways, including MQTT (Message Queuing Telemetry Transport) endpoints. The Waveshare board is supplied with an example that transmits values from the temperature sensor on the Pico. You can connect

Below ♦ You get a lot of kit for your money, including a LoRa antenna and a rechargeable battery

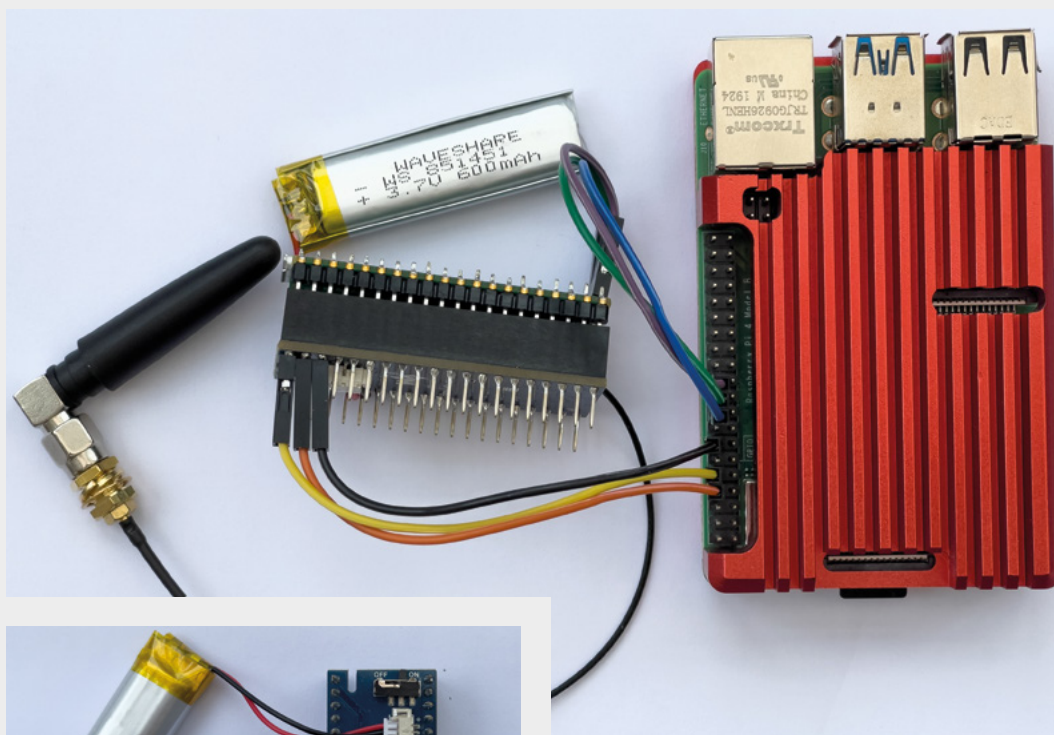
You don’t just get the device – you also get an external antenna and a rechargeable battery



more sensors to the Pico, and send their readings in LoRa packets.

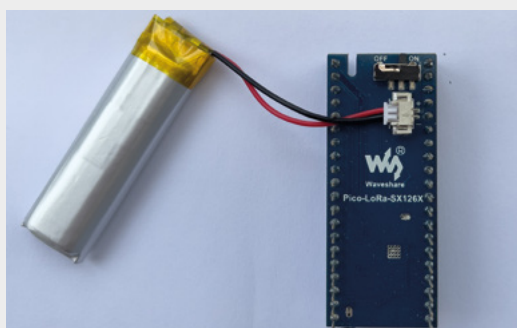
For your money, you don’t just get the device – you also get an external antenna and a rechargeable battery. You just need to add a Pico and any sensor inputs or control outputs that you want to connect to it. The hardware is especially well thought out. You plug your Pico into it, but it also provides pass-through pins to connect to other devices, or for plugging into a breadboard.

You program the device in C++ using Visual Studio Code. And this is where things start to get a bit tricky. Installing Visual Studio Code on a computer is easy enough, but configuring it to create Pico



Left You have to solder three extra pins onto the top of your Pico to connect the debug signals

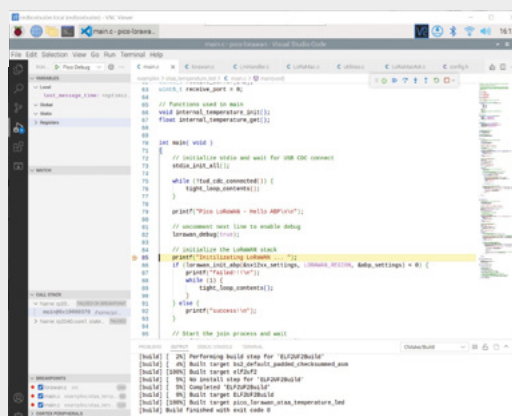
Below This shows Visual Studio running one of the sample programs that connects to The Things Network and sends temperature values to a LoRaWAN application



Above The device includes a power switch you can use to disconnect the battery power. This is a great feature; it means that you can manually power everything off without having to add power-switching code in the program that controls the device

programs turns out to be a bit trickier. The best solution is to use a Raspberry Pi as the development device. The getting started guide (at hsmag.cc/PicoGetStarted) describes how to use a batch file that sets up everything automatically. It even shows you how to set up in-circuit debugging, which turned out to be very useful when testing this device. The in-circuit debugging works by linking the Pico directly to the development device. C++ programs are compiled on the Raspberry Pi and then sent into the Pico. The additional connections on the top of the Pico are used to control the debugging process. Note that if you want to use the serial output from the Pico to interact with your programs, you can connect the USB port on the Pico to a device and then run a terminal program on that device to talk to it.

Before a device can interact with a LoRaWAN application, it must first be added to the application.



You do this on The Things Network console web page. Each device in an application has a unique address, and all messages sent by the device contain this address. Any messages the device sends are received by any gateway, and the device address is used to direct the message to the corresponding applications. And here we hit a snag. Whenever our device tries to connect to LoRaWAN, it uses an incorrect device address.

Fortunately, we traced the problem to a small part of the driver that converts the addresses from text into the numeric values used in the transmission. Thanks to Brian N for his help with this. At the time of writing, Waveshare is aware of the problem and an updated version of its software will be available soon.

VERDICT

This board would be a great way to get started with LoRaWAN using the Raspberry Pi Pico. You get an awful lot for your money.

9/10

issue
#58

ON SALE
18 AUGUST

HACK YOUR FOOD

Also

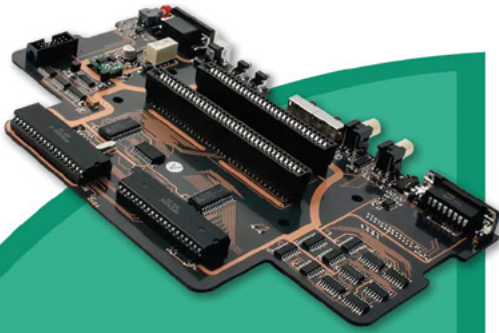
- PICO W
- 3D PRINTING
- LASER CUTTING
- ELECTRONICS
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

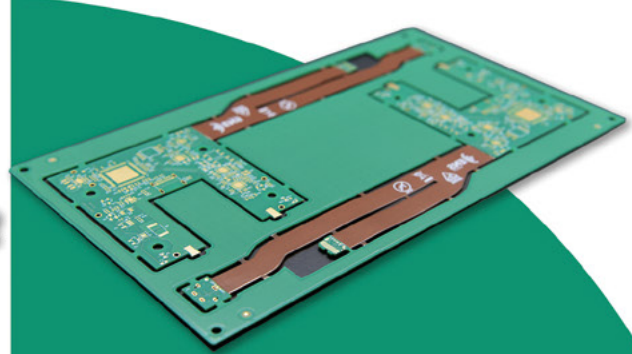


next month



- High-Quality PCBA with component sourcing
- Start from only **\$30**
- Free stencil & Free shipping all over the world

Turnkey PCB Assembly



- Instant quote online & User-friendly web interface
- Fast turnaround in **24 hours**
- Make your unique and high-end products the easy way

Advanced PCB Fabrication

**Quality
Guaranteed**

3D Printing & CNC Machining

- Custom metal & plastic parts for **only \$25**
- On-demand production and rapid prototyping in **as fast as 1 day** leading the digital manufacturing revolution.



More Custom Options

- Professional Flex/Rigid-Flex PCB
- New soldermask colors: **Pink, Gray, Orange and Transparent**
- Perfectly implement your idea with different types of PCB

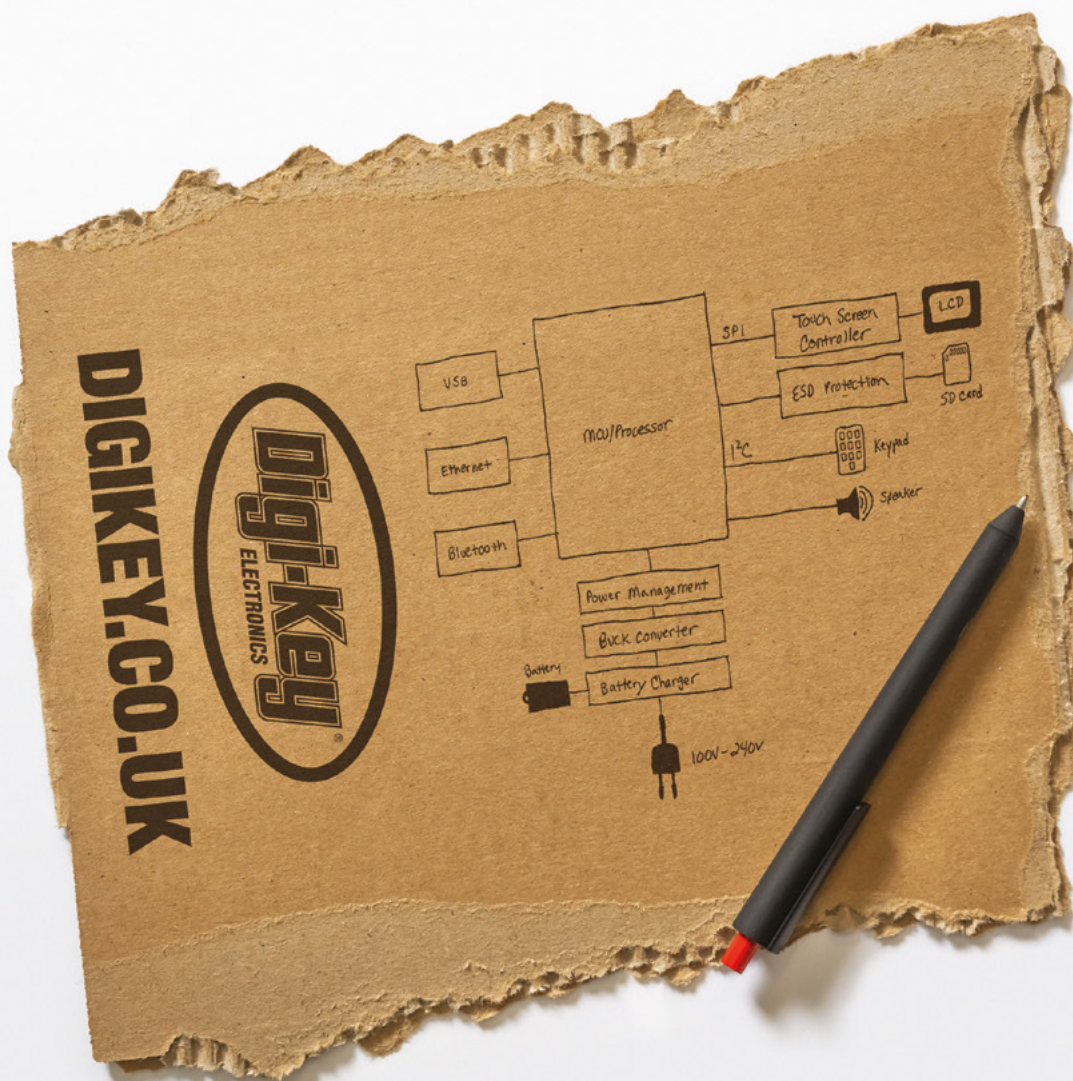


More information:
www.PCBWay.com

Email:
service@pcbway.com

Phone:
(0571) 8531 7603

IDEAS START HERE



From millions of in-stock parts to the latest new product inventory, we've got everything you need to turn breakthrough ideas into reality.

Get inspired at [digikey.co.uk](https://www.digikey.co.uk) or call 0800 587 0991.



Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel